

Paweł Saniewski

s16996, WIs I.3 - 19c

Sobota, 1 grudnia 2018r.

Raport Projektu

1. Ogólny opis rozwiązania

Program składa się z dwóch klas - PlayerNode definiujący gracza i mechanizm gry, oraz Main rozpoczynający program i odpowiadający za prawidłowe utworzenie gracza i wejście do gry. Przebieg gry jest ściśle ukierunkowany, a wynika to z zastosowanego tekstowego interfejsu użytkownika. Przykładowe uruchomienie kilku procesów programu (fragmenty w kolorze zielonym są wprowadzane przez użytkownika):

A.

```
Type in your name: Gracz1
Type: 
'1' to start a new tournament,
'2' to join an existing tournament.
1
Pawes-MacBook-Pro.local/192.168.0.12 (Gracz1) is listening on
port: 51590
```

B.

```
Type in your name: Gracz2
Type: 
'1' to start a new tournament,
'2' to join an existing tournament.
2
Please, type in the ip you wish to connect to: 192.168.0.12
Please, type in the port you wish to connect to: 51590
```

C.

```
Type in your name: Gracz3
Type: 
'1' to start a new tournament,
'2' to join an existing tournament.
2
Please, type in the ip you wish to connect to: 192.168.0.12
Please, type in the port you wish to connect to: 51590
```

2. Szczegółowy opis rozwiązania

Przebieg gry jest następujący:

- Wpisujemy dowolną nazwę gracza.
- Wpisujemy "1", jeżeli chcemy utworzyć nowy turniej, lub "2", jeżeli chcemy dołączyć do istniejącego turnieju.

Krok	Jeżeli wybraliśmy 1:	Jeżeli wybraliśmy 2:
1.	Otrzymujemy komunikat o otwarciu portu o numerze przydzielonym automatycznie pod adresem IP odpowiadającym adresowi lokalnemu (127.0.0.1).	Wpisujemy adres IP i port gracza rozgrywającego turniej.
2.	Czekamy na komunikat od gracza.	Dołączamy do gry łącząc się z graczem, którego adres IP i port podaliśmy przed chwilą i wysyłamy komunikat "Contacts".
3.	Po otrzymaniu komunikatu "Contacts" do nadawcy komunikatu wysłane zostają dane kontaktowe o wszystkich graczach w turnieju.	Otrzymujemy dane kontaktowe wszystkich graczy, a następnie z każdym z graczy rozgrywamy pojedynek łącząc się z nim i wysyłając komunikat "Play" razem z wprowadzoną przez nas liczbą.
4.	Po otrzymaniu komunikatu "Play" wprowadzamy liczbę, którą chcemy zagrać. Dopiero po wprowadzeniu naszej liczby możemy wczytać liczbę otrzymaną z komunikatem. Odliczanie rozpoczynamy od nadawcy komunikatu, a następnie wynik rozgrywki zostaje zapisany w pamięci i wysłany do przeciwnika.	Od każdego gracza otrzymujemy wynik, po czym zostaje on zapisany w pamięci rozgrywek. Następnie wszystkim graczom wysyłamy komunikat "NewContact", który nakazuje dodanie nas do listy kontaktów adresata.
5.	Po otrzymaniu komunikatu "NewContact" do listy kontaktów dodajemy informacje o nadawcy.	Do własnej listy kontaktów dodajemy informacje o sobie, a następnie otwieramy nowy port nasłuchowy i czekamy na komunikaty od nowych graczy.

- Jeżeli chcemy wyjść z rozgrywki, zwyczajnie przerywamy proces programu. Przerwanie procesu automatycznie wysyła do wszystkich graczy komunikat "Quit", który powoduje usunięcie wszystkich informacji kontaktowych i wyników rozgrywek.
- Turniej może się toczyć dopóki pozostanie w nim jeden gracz.
- Nowi gracze mogą łączyć się z dowolnymi graczami, którzy są uczestnikami turnieju.

Każdy gracz przechowuje następujące informacje:

- Swoją nazwę, którą przedstawia się podczas rozgrywek,
- Automatycznie przydzielony numer portu komunikacyjno-nasłuchowego, który jednocześnie jest jego jednoznacznym identyfikatorem,
- Adres IP agenta wprowadzającego - jest to adres gracza, z którym połączono się po raz pierwszy,
- Numer portu agenta wprowadzającego - jest to numer portu, do którego połączono się po raz pierwszy,
- Listę agentów aktualnie w sieci,
- Kolekcję przechowującą zestawy (agent - wynik)

3. Obserwacje, eksperymenty, wnioski i założenia

Do rozwiązania dołączony jest plik źródłowy GUI.java, który jest niedokończonym fragmentem projektu. Koncepcją było utworzenie przejrzystego i łatwego w obsłudze graficznego interfejsu, który pozwalałby na pełną kontrolę agenta, wgląd do tablicy wyników rozgrywek, stały dostęp do wszystkich agentów w sieci oraz samodzielne wysyłanie komunikatów za pomocą wybrania agenta z sieci i naciśnięcia przycisku odpowiadającemu komunikatowi.

W rozwiązaniu projektu przyjęto zgodność typów wprowadzanych przez użytkownika z oczekiwanymi przez program. Niektóre nazwy w kodzie odbiegają od tych podanych w opisie projektu np. komunikat JOIN i przyjęty przeze mnie komunikat Contacts. Wynika to z zachowania jednoznaczności nazw i pełnionych przez nie funkcje.