

# Protokół sieciowy

## Opis

Protokół zaimplementowany w naszej aplikacji jest protokołem tekstowym – wszystkie dane wysyłane są w formie tekstu i później odpowiednio odczytywane i parsowane na odpowiednie typy po stronie klienta. Serwer ma możliwość obsługi wielu klientów (każdy klient obsługiwany jest w osobnym wątku). Serwer wysyła do klienta dane konfiguracyjne, dane poszczególnych poziomów oraz listę najlepszych wyników. Klient wysyła do serwera wynik po wygraniu gry.

C – Klient

S - Server

## Pobieranie danych konfiguracyjnych

Klient w celu pobrania danych konfiguracyjnych wysyła do serwera informację, że chce je uzyskać:

C: Get config data=> S

Na co serwer odpowiada wysłaniem danych (które były wcześniej pobrane z pliku i są po stronie serwera przechowywane w liście w typie String) w formacie: #DANA dana, czyli

S: #DANA dana => C

A po wysłaniu ostatniej danej serwer wysyła komunikat:

S: Config Data sent => C

Co po stronie klienta jest informacją o tym, że wszystkie dane zostały wysłane i należy przerwać odbieranie danych.

Przykład:

```
Client 2: Get config data
Server: #WIDTH
Server: 667
Server: #GAMEHEIGHT
Server: 550
Server: #OBJECTDIM
Server: 50
Server: #BARHEIGHT
Server: 80
```

```
Server: #LEVELSTRINGY
Server: 23
Server: #NICKSTRINGX
Server: 325
Server: #NICKSTRINGY
Server: 23
Server: Config Data sent
```

## Pobieranie danych poziomu

Klient w celu pobrania danych konkretnego poziomu wysyła do serwera informację, że chce je uzyskać:

C: Get level data(numer poziomu)=> S

Na co serwer odpowiada wystaniem danych (pobranych wcześniej z pliku i które po stronie serwera przechowywane są w liście w typie String) w formacie #OBIEKT x y velx (opcjonalnie) vely (opcjonalnie),czyli:

S: #OBIEKT x y velx vely => C

Po wysłaniu danych o ostatnim obiekcie w liście serwer wysyła komunikat:

S: Level data sent => C

Co podobnie jak przy pobieraniu danych konfiguracyjnych jest sygnałem dla klienta, że wszystkie dane zostały wysłane i należy przerwać ich odbieranie.

Przykład:

```
Client 2: Get level data1
Server: #BONUSPOINTGOLD
Server: 0
Server: 300
Server: #BONUSPOINTGOLD
Server: 0
Server: 350
```

```
Server: #COBSTACLE
Server: 550
Server: 250
Server: Level data sent
```

## Pobieranie listy najlepszych wyników

W celu pobrania listy najlepszych wyników klient wysyła do serwera informację, że chce ją uzyskać:

C: Get ranking table => S

Na co serwer odpowiada wystaniem listy najlepszych wyników pobranej wcześniej z pliku i przechowywanej po stronie serwera w liście w typie User (którego obiekty zawierają informację o nicku i wyniku konkretnego użytkownika – te informacje wysyłane są w formie tekstowej do klienta)

S: Nick Wynik => C

Po wysłaniu wszystkich danych, serwer komunikuje o tym klienta wysyłając informację:

S: Ranking table sent => C

Co jak wcześniej – jest informacją dla klienta o tym, że wszystko zostało przesłane przez serwer i należy już zakończyć odbieranie danych.

Przykład:

```
Client 2: Get ranking table
Server: Monika 5800
Server: Mb 4600
Server: Gosia 4400
Server: Kasia 4100
Server: Karo 3850
Server: Basia 2100
Server: Ranking table sent
```

## Przesłanie wyniku na serwer

W celu przesłania wyniku na serwer najpierw klient wysyła do serwera informację o tym (w celu umożliwienia odebrania go przez serwer):

C: Sending score => S

Po tym klient wysyła dane Usera (nick i wynik)

C: Nick wynik => S

Po odebraniu tych danych, serwer wysyła informację do klienta o pomyślnym odebraniu danych:

S: Score received => C

Przykład:

```
Client 1: Sending score
Client 1: Imię
Client 1: 3000
Server: Score received
```