Internet, 5.05.2020r.

Maciej Krysiak
Dominik Baczyński

Protokół sieciowy dla gry Lunar Lander

Zadaniem przedstawionego poniżej protokołu sieciowego będzie odpowiadał za nawiązywanie połączenia pomiędzy grą Lunar Lander, a serwerem, w celu pobierania danych z plików konfiguracyjnych gry znajdujących się na serwerze, wczytywania poziomów gry, pobieraniem listy najlepszych graczy i zapisywaniem najnowszy wyników do pliku. W protokole zostało uwzględnione również polecenie GET i PUT do pobierania i zapisywania dowolnych danych z plików serwera.

Jest to protokół tekstowy. Do jego implementacji wykorzystane są standardowe klasy Javy, takie jak Socket oraz podstawowe protokoły transportowe, takie jak TCP. Połączenie nie jest w żaden sposób szyfrowane. Komunikacja odbywa się na podstawie wymiany jednej linii tekstu zakończonej znakiem nowej linii. W zależności od żądania, pakiety danych mogą mieć różną długość.

KOMENDY PROTOKOŁU SIECIOWEGO

- **Logowanie się na serwerze**

  W celu zalogowania klienta do serwera, klient wysyła żądanie:

  ```
  K: LOGIN\n -> S
  ```

  Serwer w odpowiedzi może zalogować użytkownika, wysyłając odpowiedź:

  ```
  S: LOG_IN (nr_klienta)\n -> K
  ```

  Lub może odmówić zalogowania użytkownika, odsyłając wiadomość:

  ```
  S: CONNECTION_REJECTED\n -> K
  ```

- **Wylogowanie się z serwera**

  W celu wylogowania użytkownika z serwera, klient wysyła żądanie:

  ```
  K: LOGOUT\n -> K
  ```

  W odpowiedzi serwer zamyka połączenie i odsyła do klienta wiadomość:

  ```
  S: LOGOUT\n -> K
  ```

- **Pobieranie ustawień gry**

  W celu pobrania ogólnych ustawień gry, klient wysyła żądanie:

  ```
  K: GAME_SETTINGS\n -> S
  ```

  Serwer wysyła w odpowiedzi listę parametrów gry wraz z ich wartościami:

  ```
  S: klucz#wartość@klucz#wartość@klucz#wartość\n -> K
  ```

- **Pobieranie parametrów o wyglądzie ekranu menu**

  W celu pobrania parametru wyglądu menu, klient wysyła żądanie:

  ```
  K: GET_MENU_SETTINGS\n -> S
  ```

  Serwer wysyła w odpowiedzi listę parametrów wyglądu menu wraz z ich wartościami:

  ```
  S: klucz#wartość@klucz#wartość@klucz#wartość\n -> K
  ```

- **Pobieranie tekstu pomocy gry**

  W celu pobrania tekstu pomocy gry (zasady, sterowanie), klient wysyła żądanie:

  ```
  K: GET_HELPTEXT\n -> S
  ```

  Serwer wysyła w odpowiedzi linie tekstu pomocy:

  ```
  S: text1#wartość@text2#wartość@text3#wartość.\n -> K
  ```

- **Pobieranie mapy (należy wskazać poziom, który chce się pobrać)**

  W celu pobrania poziomu gry z serwera, klient wysyła żądanie:

  ```
  K: LOAD_LEVEL:1\n -> S
  ```

  Serwer wysyła w odpowiedzi poziom gry:

  ```
  S: klucz#wartość@klucz#wartość@klucz#wartość\n -> K
  ```

- **Pobranie listy wyników najlepszych graczy**

  W celu pobrania listy najlepszych wyników, klient wysyła żądanie:

  ```
  K: GET_SCOREBOARD\n -> S
  ```

  Serwer wysyła w odpowiedz listę najlepszych wyników graczy w postaci dwóch list - listy nicków graczy i listy ich wyników:

  ```
  S:  nicks#a,b,c,d,e@scores#5,4,3,2,1\n -> K
  ```

- **Przekazywanie najlepszych wyników na serwer:**

  W celu przesłania wyniku gracza do tablicy wyników, klient wysyła żądanie:

  ```
  K: SAVE_SCORES:nicks#PlayerNick@PlayerScore\n -> S
  ```

  - Jeśli wynik gracza kwalifikuje się do zapisania na liście, serwer odpowiada komunikatem:

    ```
    S: SCORE_SAVED\n -> K
    ```

  - Jeśli wynik gracza NIE kwalifikuje się do zapisania na liście, serwer odpowiada komunikatem:

    ```
    S: SCORE_TOO_LOW\n -> K
    ```

- **Pobieranie dowolnej liczby danych z konkretnego pliku konfiguracyjnego (trzeba znać nazwę pliku)**

  W celu pobrania określonych danych, klient wysyła następującej postaci żądanie:

  - Jeden klucz:

    ```
    K: GET:nazwa_pliku@klucz\n -> S
    ```

  - Wiele kluczy:

    ```
    K: GET:nazwa_pliku@klucz@klucz@klucz\n -> S
    ```

  Server wysyła w odpowiedzi listę danych:

  - Dla jednego klucza:

    ```
    K: klucz#wartość -> klucz#wartość
    ```

  - Dla wielu kluczy:

    ```
    K: klucz#wartość@klucz#wartość@klucz#wartość\n -> S
    ```

- **Umieszczenie w plikach konfiguracyjnych serwera dowolnej liczby wartość (trzeba znać nazwę pliku)**

  W celu umieszczenia określonych danych w plikach konfiguracyjnych, klient wysyła żądanie:

  - Dla jednego klucza

    ```
    K: PUT:nawa_pliku@klucz#wartość \n-> S
    ```

  - Dla wielu kluczy:

    ```
    K: PUT:nawa_pliku@klucz#wartość@klucz#wartość \n-> S
    ```

  Server odpowiada sumą kontrolną:

  - Dla jednego klucza

    ```
    S:  VALUE_SAVED\n -> K
    ```

- **`FATAL_ERROR` – każde żądanie może skutkować taką odpowiedzią, jeśli żądanie klienta jest obsługiwane, lecz podczas jego wykonywania napotkano problem**

  Odpowiedź serwera:

  ```
  S: FATAL_ERROR\n -> K
  ```

- **`INVALID_COMMAND` – odpowiedź serwera, w przypadku otrzymania żądania nieobsługiwanego przez serwer**

  Odpowiedź serwera:

  ```
  S: INVALID_COMMAND\n ->K
  ```