

Protokół sieciowy

Realizowany protokół sieciowy opisuje relacje między klientem aplikacji a serwerem. Klient będzie mógł pozyskać dane, które przechowuje i do których dostęp ma serwer. Będzie możliwe pobranie z serwera następujących danych: lista najlepszych wyników, pliki konfiguracyjne aplikacji, dane konfiguracyjne poziomów gry (w zależności, ile tych poziomów będzie na serwerze). Dodatkowo klient będzie mógł wysłać do serwera aktualny wynik rozgrywki, który zostanie dodany do listy najlepszych wyników(lub nie, serwer zdecyduje). Użyjemy protokołu tekstowego, przesyłanego za pomocą TCP.

Żądania oraz odpowiedzi klienta i serwera:

(C-klient, S-serwer, =>kierunek komunikacji, \n – znak końca wiadomości)

- Pobieranie danych z serwera:

C:GET_HIGHSCORES\n=>S

Klient nadaje prośbę o wysłanie listy najlepszych wyników.

S:HIGSCORES: Andrzej_1000000/polandball_574/Maciek_175\n=>C;

Serwer wysyła dane dotyczące listy najlepszych wyników gry. Znak "/" oddziela każdego gracza wraz z wynikiem, natomiast znak "_" konkretny wynik od gracza. Klient obrabia o trzymane dane.

C:GET_CONFIGFILE\n=>S;

Klient nadaje prośbę o wysłanie danych konfiguracyjnych aplikacji.

S:CONFIGFILE: 700 1400 800 800 500 5 1 20 100 10 1000 50 500 3\n=>C;

Serwer wysyła dane konfiguracyjne gry, znaczenie poszczególnych liczb oddzielonych spacją zostało opisane w dokumentacji klasy ParseConfigFile (np. Liczba 700 oznacza zawsze wysokość okna gry).

C:GET_LEVEL: <numer_poziomu>\n=>S;

Klient nadaje prośbę o wysłanie danych konfiguracyjnych konkretnego poziomu (numer_poziomu: typ int > 0 np. 1,2,3..itd.).

S:LEVEL <numer_poziomu>: 2 2 3 2 50 50 1 1 1 760&N_ NG%N_ N_\n=>C;
lub

S:LEVEL_NOT_FOUND\n=>C;

Serwer wysyła dane konfiguracyjne dotyczące żadanego poziomu(numer_poziomu – informacja od klienta). W wiadomości można wyróżnić 2 części (oddzielone znakiem "&") pierwsza dotyczy oddzielonych spacją parametrów poziomu (np. pierwsza liczba 2 oznacza ilość kolumn danego poziomu), druga to zakodowane pola planszy. Szczegółowe informacje dotyczące parametrów poziomu oraz ich kolejności zostały zawarte w dokumentacji klasy ParseLevelFile.

W przypadku braku danych dotyczących żadanego poziomu serwer wysyła informację, że nie może znaleźć danego poziomu. Klient po otrzymaniu takiej informacji kończy daną rozgrywkę.

- Aktualizacja listy najlepszych wyników gry:

C:PUT_SCORE <nazwa_gracza> <wynik>\n=>S;

Klient wysyła do serwera nazwę gracza (typ: String, nie są dozwolone spacje – maksymalna długość nazwy gracza: 10) oraz jego wynik (typ: int) w danej rozgrywce, serwer zapisuje go do swojego pliku highscores (lub nie).

S:SCORE_RESPONSE: <decision>\n =>C;

Serwer wysyła informację o dodaniu lub niedodaniu wyniku do listy (zmienna decision typu boolean: true, gdy wynik i nazwa gracza zostały dodane do listy najlepszych wyników, false, gdy nie). Serwer obrabia otrzymane dane.

- Inne komendy:

S:INVALID_COMMAND\n =>C;

Serwer wysyła do klienta informację o nieznanej komendzie, gdy nie potrafi odpowiedzieć na żądanie klienta. Klient nie odpowiada na tą wiadomość.