

PROZE – gra Lunar Lander, etap IV

Protokół sieciowy komunikacji z serwerem aplikacji

Funkcjonalność sieciowa programu „Lunar Lander” obejmuje serwer przechowujący i udostępniający informacje konfiguracyjne (definicje wyglądu plansz, scenariusz gry), który w osobnych wątkach obsługuje dowolną liczbę klientów. Może on również zapisywać wyniki graczy do rankingu, bądź je stamtąd wczytywać i umożliwić ich wyświetlenie. Serwer charakteryzujemy jako niezależną, bezobsługową, parametryzowaną aplikację. Oczywiście istnieje możliwość, by aplikacja w wersji sieciowej działała także bez serwera. Wówczas użytkownik po prostu otrzyma niezbędne dane z lokalnych plików konfiguracyjnych, dołączonych do programu w postaci plików tekstowych. Adres serwera i numer portu są parametrami aplikacji klienta i są wyświetlane po uruchomieniu.

Aplikacja kliencka w przypadku wyrażenia chęci zainicjowania połączenia z serwerem wymagać będzie od użytkownika zalogowania się poprzez podanie adresu IP serwera oraz numeru portu. Po prawidłowym zalogowaniu klient będzie mógł wysłać żądanie o udostępnienie niezbędnych danych pochodzących z plików konfiguracyjnych, a po zakończeniu rozgrywki również poprosić o zapisanie swojego wyniku końcowego, by móc go potem porównać z osiągnięciami innych graczy.

Sam protokół będzie miał typ tekstowy, zaś pakiet przesyłu informacji będzie stanowiła pojedyncza linia tekstu (‘\n’). Parametry zdań protokołu mogą być różnego typu, a jeśli w jednym zdaniu znajdować się będzie kilka parametrów, wówczas będą one odseparowane za pomocą myślników (‘-’).

Przykładowa wymiana żądań klienta (C) i odpowiedzi serwera (S) została

przedstawiona poniżej (z racji na fakt, że „serwer (...) w osobnych wątkach obsługuje dowolną liczbę klientów” serwer nadaje klientom identyfikatory, aby móc odróżnić później, kto jest nadawcą danego żądania; i-ty klient oznaczony jest poniżej jako „C_i”):

- Klient nawiązuje kontakt z serwerem (loguje się):
C: **login** => S
- Serwer potwierdza zalogowanie klienta, nadaje mu identyfikator ‘i’ (int):
S: **loggedin** i => C_i
- Klient prosi o udostępnienie definicji wyglądu planszy oraz scenariusza gry:
C_i: **getPropertiesGry** => S
- Serwer wysyła klientowi definicje wyglądu planszy (int) oraz scenariusz gry (int):
S: **givePropertiesGry** xWierzcholkiPlanety-yWierzcholkiPlanety-xLadowiska-yLadowiska-xStartoweStatku-yStartoweStatku-xBonusu-yBonusu-xPredkoscMax-yPredkoscMax-xPredkoscLadowania-yPredkoscLadowania-liczbaStatkow-paliwo-liczbaPoziomow-punktyZaPlansze-punktyZaBonus => C_i
- Klient wysyła serwerowi swój nick ‘*nick*’ (String) oraz wynik końcowy ‘*wynik*’ (int):
C_i: **sendScore** nick-wynik => S
- Serwer potwierdza (bądź nie) ‘*czyPotwierdzono*’ (boolean) zapisanie wyniku gracza:
S: **saveScore** czyPotwierdzono => C_i
- Klient prosi serwer o wyświetlenie rankingu najlepszych graczy:
C_i: **showRanking** => S
- Serwer udostępnia klientowi ranking najlepszych graczy w postaci mapy z parami parametrów: miejsce w rankingu ‘*miejsce*’ (int) oraz nick gracza ‘*nick*’ (String):
S: **displayRanking** miejsce-nick=> C_i