

PROZE

Dyna Blaster

Projekt protokołu sieciowego

Protokół sieciowy ma za zadanie komunikację (połączenie i wymianę informacji) pomiędzy aplikacją serwera i klienta. Ma to służyć pobieraniu danych konfiguracyjnych do gry, poziomów oraz obecnej tablicy wyników.

Protokół wykorzystuje protokół TCP, standardowe klasy i interfejsy Java. Znak nowej linii jest informacją o końcu przesyłanych danych.

W opisie protokołu przyjmujemy skróty:

K – informacja wysłana przez klienta

S – informacja wysłana przez serwer

1. Logowanie i nawiązywanie połączenia:

Po nawiązaniu połączenia przez protokół TCP klient wysyła żądanie logowania:

K: LOGIN\n

Serwer sprawdza dane logowania i zwraca odpowiedź:

a) dane poprawne:

S: LOGIN_DONE\n

b) dane niepoprawne:

S: LOGIN_FAILED\n

2. Pobieranie plików konfiguracyjnych:

W celu pobrania plików konfiguracyjnych, które zawierają informacje o różnych parametrach rozgrywki, klient wysyła żądanie o pobranie tychże plików:

K: GET_SETTINGS\n -> S

Serwer po otrzymaniu żądania, przesyła ustawienia gry:

S: <?xml version="1.0" encoding="UTF-8"?>

<config>

```
<LEVEL_WIDTH>15</LEVEL_WIDTH>
```

```
<LEVEL_HEIGHT>15</LEVEL_HEIGHT>
```

```
[...]
```

```
</config>\n
```

Po odebraniu ustawień, są one obrabiane i przyporządkowywane odpowiednim zmiennym w programie.

3. Pobranie listy najlepszych wyników:

K: GET_HIGH_SCORES\n

Serwer po odebraniu takiego żądania odsyła:

S: HIGH_SCORES: ?xml version="1.0" encoding="UTF-8"?>

```
<highscores>
```

```
  <ScoreID id=NUM><name>nazwa</name><score>wartość</score></ScoreID>
```

```
  [...]
```

```
</highscores>\n
```

4. Pobranie informacji o dostępnych poziomach:

K: GET_NUMBER_OF_AVAILABLE_LVLIS\n

Odpowiedź serwera:

S: NUMBER_OF_AVAILABLE_LVLIS: FIRST: 1; LAST: NUM\n

5. Pobranie poziomu:

W celu pobrania danych poziomu, który będzie aktualnie rozgrywany, klient wysyła żądanie do serwera:

K: GET_LEVEL_X\n -> S

Po otrzymaniu żądania, serwer przesyła dane poziomu do klienta:

S: LEVEL: 1: <?xml version="1.0" encoding="UTF-8"?>level<rowID="1"><column ID="1">U

```
</column><column ID="2">U</column><column ID="3">U</column>
```

```
[...]
```

```
</row>
```

```
[...]
```

```
</level>\n
```

Po otrzymaniu informacji o poziomie, są one odpowiednio obrabiane i przygotowywana jest z nich plansza do gry. Klient natomiast wysyła odpowiedź, że pliki zostały poprawnie odebrane.

6. Wysłanie wyniku do tablicy najlepszych wyników:

K: UPDATE_HIGH_SCORES:*Nazwa_wynik*\n

Serwer sprawdza czy uzyskany wynik należy dodać do najlepszych wyników i odpowiada:

a) Jeśli tak:

S: HIGH_SCORE_ACCEPTED:*NUM_pozycja_na_liście*\n

b) Jeśli nie:

S: HIGH_SCORE_REJECTED\n

7. Zakończenie połączenia:

K: LOGOUT\n

Potwierdzenie serwera o wylogowaniu:

S: LOGOUT_DONE\n

Przy przerwaniu połączenia na życzenie serwera:

S: CLOSE_CONNECTION\n

Odpowiedź klienta:

K: CLOSE_CONNECTION_DONE\n