

# *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.

#### 1. Dane ogólne protokołu:

Protokół wykorzystuje protokół TCP, standardowe klasy i interfejsy Java.

Typ: **tekstowy**

Porcja przesyłu: **linia tekstu** (zakończona znakiem końca linii \n)

Parametry separowane: **spacjami**.

W opisie protokołu przyjmujemy skróty:

K – informacja wysłana przez klienta

S – informacja wysłana przez serwer

#### 2. Nawiązanie połączenia:

Odbywa się przez protokół TCP.

#### 3. 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 -> S

Serwer po otrzymaniu żądania, przesyła informacje o poziomach trudności. Przesyłany jest współczynnik punktów za przeciwnika 'p' (int), współczynnik punktów za czas 't' (int), czas na przejście poziomu 'g' (int), współczynnik szybkości przeciwników 's' (int), ilość żyć gracza 'l' (int) oraz kolor tła 'c' (char):

S: p t g s l c -> K

#### 4. Pobranie listy najlepszych wyników:

K: GET\_HIGH\_SCORES -> S

Serwer po odebraniu takiego żądania odsyła serię najlepszych wyników, gdzie każde kolejne 3 parametry oznaczają jeden wynik. Każdy wynik składa się z indeksu w tabeli wyników *i* (int), nazwy gracza (String) oraz wyniku *s* (int):

S: i1 n1 s1 i2 n2 s2 i3 n3 s3 [...] i10 n10 s10 -> K

#### 5. Pobranie informacji o dostępnych poziomach:

K: GET\_NUMBER\_OF\_AVAILABLE\_LVL -> S

Odpowiedź serwera to numer najwyższego poziomu *l* (int):

S: l -> K

#### 6. Pobranie poziomu:

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

K: GET\_LEVEL\_X -> S

Po otrzymaniu żądania, serwer przesyła dane poziomu do klienta. Przesyła informację o szerokości planszy 'w' (int) oraz wysokości planszy 'h' (int), a także linijki tekstu 'r1', 'r2', ... (string), które zawierają informacje o rozkładzie pól w pojedynczym wierszu (r1 – pierwszy wiersz, r2 – drugi wiersz, ...). Przykładowe r = "UBUEUEUBUBUEUEU".

S: w h r1 r2 [...] r15 -> K

Po otrzymaniu informacji o układzie poziomu (każda litera odpowiada jednemu blokowi na planszy startowej poziomu, np. U – niezniszczalna ściana, B – kraty do zniszczenia, E – puste pole, ...) informacje są obrabiane przez odpowiednią klasę konfiguracyjną (która w zależności od tego jaki znak na danej pozycji się znajduje, przypisuje odpowiadającą komórce mapy pasujące ID). Jeżeli wszystkie informacje zostały pobrane bez problemu, klient wysyła odpowiedź, że pliki zostały poprawnie odebrane.

#### 7. Wysłanie wyniku do tablicy najlepszych wyników:

W celu zapisania wyniku na liście najlepszych wyników, klient wysyła żądanie zaktualizowania listy najlepszych wyników z dwoma danymi: nazwą gracza *n* (String) oraz jego wynikiem *s* (int):

K: UPDATE\_HIGH\_SCORES: n s -> S

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

a) Jeśli tak:

to zwraca klientowi informację na jakiej pozycji znalazł się przesłany wynik *i* (int):

S: HIGH\_SCORE\_ACCEPTED: i -> K

Jeżeli nowy wynik kwalifikuje się do tablicy najwyższych wyników, a aktualna tablica wyników posiada już 10 wyników to ostatni wynik w tablicy jest usuwany.

b) Jeśli nie:

Wysła komunikat:

S: HIGH\_SCORE\_REJECTED -> K