

4. Etap projektu z PROZE

Protokół sieciowy

Aby umożliwić uruchomienie gry na innym urządzeniu bądź na kilku naraz (obsługa wielowątkowa), koniecznym jest stworzenie dodatkowej aplikacji (oprócz samego Bombermana), z której to gra będzie pobierała wszystkie potrzebne jej dane, zamiast korzystać z plików lokalnych.

Takiego rodzaju komunikacja (zwana klient-serwer) wymaga ustalenia pewnego szablonu, według którego dane będą przesyłane. Ze względu na naturę takiego połączenia (strumień bitów) niezbędne jest usystematyzowanie, co każdy bit oznacza. Takim szablonem jest właśnie protokół sieciowy.

Komunikację klienta z serwerem można porównać do wywiadu: klient wysyła zapytania, a serwer, ustawiony po uruchomieniu na oczekiwanie na nie, odsyła wymaganą odpowiedź, na przykład dane z pliku konfiguracyjnego.

Komunikacja

Przed wysłaniem przez klienta każdego żądania nawiązane zostaje połączenie, które po odebraniu odpowiedzi przez klienta zostaje zerwane.

Skonstruowane zostały następujące polecenia:

1. Pobranie mapy poziomu.

Klient wysyła żądanie poziomu, który gracz ma przejść:

K: getLevel\n => S

Server odsyła wygląd mapy w postaci jednego Stringa który zawiera sekwencję następujących znaków(rzędy mapy oddzielone są myślnikami):

- #
- *
- P
- C
- \$
- „ ”

S: #####-#*# #-#C#P#-# \$#-#####\n => K

2. Pobranie parametrów rozgrywki.

- Klient wysyła żądanie parametrów rozgrywki:

K: getGameParameters => S

- Serwer odsyła parametry gry będące Stringami oddzielone myślnikami:

S: CreeperSpeed-CreeperPoints-PPackPoints-PlayerLives-WallPoints\n => K

3. Pobranie tekstu instrukcji.

- Klient wysyła żądanie tekstu instrukcji:

K: getInstruction\n => S

- Serwer odsyła tekst instrukcji w postaci jednego Stringa

S: tekstInstrukcji(String)\n => K

4. Pobranie rankingu.

- Klient wysyła żądanie wysłania rankingu graczy:
K: getRanking\n =>S
- Serwer odsyła ciąg bloków nick,wynik. Oba z tych parametrów to Stringi.
S: nick(String),wynik(String)-nick,wynik-nick,wynik\n

5. Zapisanie wyniku gracza.

- Klient wysyła żądanie zapisania wyniku na serwerze po zakończeniu rozgrywki:
K: saveScore: nick(String),wynik(String)\n => S
Wysłany zostaje nick gracza oraz punkty przekonwertowane na obiekt klasy String, a serwer decyduje czy wynik jest dostatecznie wysoki by znaleźć się na liście najlepszych.