Dobble

Multiplayer game

Oleksandr Mertsalov

20 sierpnia 2019

Opis programu

- Klienci - łączą się z serwerem , wysylają pakiet typu «HANDSHAKE», podłączają się do wolnego pokoju, czekają aż póki on się nie zapełni, grają razem.
- Serwer — zarządza grami klientów, w danej chwili może być prowadzonych kilka gier.

Mechanizmy:

- Zarówno klient jak i serwer, określają protokół przesyłu danych
- Serwer przez cały czas swojego działania zapisuje do logów informacje o aktualnym zdarzeniu (połączył się klient - z jakim IP, portem, …)
- Serwer i klient powinni obsługują protokół IPv4 oraz IPv6
- Zaimplementowana logika, wykorzystuje więcej niż 3 rodzaje przesyłanych wiadomości
- Synchronizacja wątków
- Serwer obsługuje 2+ graczy jednocześnie

Opis protokołu

1.1 Ogólny format pakietów

Protokól «*DBLgame*» użuwany w komunikacji między serwerem, a klientem pochodzi z rodziny protokolów «*DBLprot*» przyznaczonej dla tej gry. DBLprot korzysta z protokołu TCP/IP.

Wszystkie teksty są kodowane przy użyciu zestawu znaków UTF-8.Przy opisie struktur, założono, że char ma rozmiar 1 bajtu ,a int 4 bajtów.

Każdy pakiet zawiera :

1. Trzy stałe bajty, które oznaczają początek wiadomości <0x64,0x6f,0x62>.
2. Cztery bajty, reprezentujące rozmiar pakietu(czyli int).Maksymalny rozmiar, to 166kb.
3. «*Header*», zajmuje 1bajt i odpowiada za ogólne przyznaczenie pakietu.
4. «*Field*», zajmuje 1bajt i odpowiada za szczególowe przyznaczenie pakietu.
5. Dane.(*W zależności od pola «Field» mogą być reprzezentowane w róznych formatach.*)
6. Trzy stałe bajty które oznaczają koniec wiadomości <0x62,0x6c,0x65>.

---

* Jeśli połączyć pierwsze 3 bajty i ostatnie 3, to dostaniemy slowo dobble(nazwa gry).

1.2 Możliwe wartości dla pól Header i Field

DBLprot:

OPTION : PAKIET_DATA(0x10), REQUEST(0x30), RESPONSE(0x31).

FIELDS : ERROR(0x05)

OPTION : HANDSHAKE(0x20)

FIELDS: CONNECT(0x21),ERROR(0x05)

DBLgame:

OPTION : ROOM(0x40)

FIELDS : JOIN(0x41), LEAVE(0x42),
NEW_PLAYER(0x43),PLAYER_LEAVE(0x44),ERROR(0x05)

OPTION : GAME(0x50)

FIELD : START(0x52),CLIENT_CARD(0x53),SERVER_CARD(0x54),ANSWER(0x55),

ANSWERED(0x56),END(0x57),ERROR(0x05)

2.1 Option HANDHSAKE

Pierwszy pakiet który serwer może odebrać od klienta jest pakiet typu «*Handshake*».
Przykładowy pakiet wyglądą następująco:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → HANDSHAKE
8 → CONNECT
9 → protocol_id
10 → prtocol_version
11 → client_version
12 13 14 → 0x62,0x6c,0x65

Gdzie :

- *protocol_id* — wybór protokołu.W tym momencie «*DBLprot*» posiada tylko jeden protokól, który jest wykorzystany w grze(czyli «*DBLgame*»), przypisana mu wartość «1»;
- *protocol_version* — wersja protokołu używanego przez klienta. «*DBLgame*» jest w wersji «1.0». W postaci bajtowej ma wartość «0x10».
- *client_version* — wersja zainstalowanego klienta.

Jeśli serwer obsluguje taki protokól , w takiej samej wersji i klient używa najnowszej wersji programu, to odpowie:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → HANDSHAKE
8 → CONNECT
9 → SUCCESS
10,11,12 → 0x62,0x6c,0x65

W tym pakiecie, «*Dane*» , to kod odpowiedżi, który zajmuje 1 bajt.

2.2 Option ROOM

Żeby dolączyć się do pokoju,w którym będzię prowadzana gra, klient muśi wysłac pakiet typu «ROOM» :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → ROOM
8 → JOIN
9 → nickname_length
10-26 → nickname
27 28 29 → 0x62,0x6c,0x65

W typ pakiecie klient przesyła swój *nickname* , który może posiadać znaki z tablicy ASCII w przedziałie [0x20,0x7e) oraz dlugość tego pola w przedzilie [0x03,0x10].

Jeśli nickname jest poprawny i jest pokój do którego można dodać klienta serwer odpowie pakietem :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → ROOM
8 → JOIN
9 → room_nbr
10 → game_id
11 → max_players
12-? → players_info
?,?,? → 0x62,0x6c,0x65

Gdzie :

- *room_nbr* — numer pokoju do którego będzie dodany klient.
- *game_id* — id gry.Ta wartość, będzie zmieniała się, w załeżności od ilośći prowadzonych gier w pokoju .(Tzn. Pierwsza gra ma id=1,druga id=2 … i td.).
- *max_players* — ile osób maksymalnie może być podłączono do tego pokoju.
- *players_info* — informacja o podłączanych do tego pokoju klientów. Informacja jest typu String, w formacie «1klientID\r1klientNickname\n2klientID\r2klientNickname\n».Klient odejmuje od długości calego pakietu 3 ostatnie bajty ,żeby wiedzić ile bajtów zajmuję to pole.

Kiedy klient został dodany do pokoju,wszyscy uczestniki w tym pokoju otrzymają wiadomość od serwera.Przykładowo:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → ROOM
8 → NEW_PLAYER
9-? → player_info
?,?,? -> 0x62,0x6c,0x65

player_info — informacja o nowym uczestniku.Informacja jest typu String, w formacie «klientID\rklientNickname\n».

Kiedy klient opuści pokój, wszyscy uczestniki otrzymają wiadomość:

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → ROOM
8 → PLAYER_LEAVE
9 → playerID
10,11,12 → 0x62,0x6c,0x65

2.3 Option REQUEST , RESPONSE

Gra się zaczyna, kiedy w pokoju jest maksymalna liczba klientów i wszyscy są gotowi.Zeby sprawdzić ich gotowność, serwer wysyła żądanie :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → REQUEST
8 → READY
9,10,11 → 0x62,0x6c,0x65

Od uzytkownika załeży ,czy wyślę klient odpowiedż, czy nie. Jeśli tak to pakiet będzie wygłądał :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → RESPONSE
8 → READY
10,11,12 → 0x62,0x6c,0x65

W protokolach DBLprot tyłko te dwa pakiety nie mają pola dła danych.

2.4.1 Option GAME

Gra się zaczyna, gdy serwer wyśli do wszystkich użytkowników pokoju pakiet :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME

8 → START
9 → amount_of_card
10,11,12 → 0x62,0x6c,0x65

Bajt nr. 9 odpowiada za ilość kart, który serwer wydał każdemu klientu. Gra trwa do póki w pokoju nie zostanie jeden klient z kartami.

2.4.2 Proces gry

Serwer wysyła każdemu uczestniku gry, po jednej karcie:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME
8 → CLIENT_CARD
9-? → Card
?,?,? → 0x62,0x6c,0x65

W tym pakiecie, karta reprezentowana jako obiekt classy «*Card*». Każda karta w sobie zawiera numer i listę obrazków, które są na niej. Każdy obrazek, to klasa, która zawiera numer i śćieżke do obrazku na urządzeniu klienta.

Następnie serwer wysyłą do wszystkich uczestników gry, ten sam obrazek(plik) który jest kartą serwera:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → Protocol.GAME
8 → Protocol.SERVER_CARD
9-? → Card
?,?,? → 0x62,0x6c,0x65

Każdy z uczestników może przesłać swoją odpowiedź reprezentowana, jako numer obrazku:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME
8 → ANSWER
9 → imageID
10,11,12 → 0x62,0x6c,0x65

Jeśli obrazek na który wskazał klient jest wsród obrazków karty serwera to wtedy ta karta będzie zaakceptowana i serwer wyśli do wszystkich , id uczestnika który był pierwszy:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME
8 → ANSWERED
9 → playerID
10,11,12 → 0x62,0x6c,0x65

Gra się kończy wysłąniem z serwera do wszystkich uczestników pakietu :

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME
8 → END
9 → result_table
10,11,12 → 0x62,0x6c,0x65

result_table — String w formacie «id1\nid2\nid3\nid4\n».

3.0 Obsluga błędów

W przypadku gdy klient przesłal nie poprawny pakiet , serwer odpowie:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → PACKET_DATA
8 → ERROR
9 → kod_odpowiedzi
10,11,12 → 0x62,0x6c,0x65

kod_odpowiedzi:

- PACKET_LENGTH(0x16) — dlugość pakietu jest nie poprawna
- PACKAGE_SYNTAX(0x17) — pakiet nie jest standartem DBLprot
- OPTION(0x18) — nie prawidlowe pole OPTION
- FIELD(0x19) — nie prawidlowe pole FIELD

W przypadku gdy jest probłem z pakietem HANDSHAKE:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → HANDSHAKE
8 → ERROR
9 → kod_odpowiedzi
10,11,12 → 0x62,0x6c,0x65

kod_odpowiedzi:

- PROTOCOL(0x27) — klient żada komunikować używając protokól nieznany dłą serwera.
- PROTOCOL_VERSION(0x28) — klient używa wersji protokolu której serwer nie posiada.
- CLIENT_VERSION(0x29) — klient ma starszą wersją programu.

W przypadku problemów z dodawaniem klienta do pokoju:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → ROOM
8 → ERROR
9 → kod_odpowiedzi
10,11,12 → 0x62,0x6c,0x65

kod_odpowiedzi:

- NICKNAME_LENGTH(0x46) — długość nickname jest nie poprawna.
- NICKNAME_CHARACHTERS(0x47) — nickname posiada niedozwolone znaki.
- ROOMS_ARE_FULL(0x48) — nie ma możliwości dołaczyć klienta pokoju, bo wszyscy są zajęte.

W przypadku nie gotowności klienta do rozpoczecią gry:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → REQUEST
8 → ERROR
9 → NOT_READY(0x36)
10,11,12 → 0x62,0x6c,0x65

W przypadku gdy gra się nie zaczeła:

nr. bajtu

0 1 2 → 0x64,0x6f,0x62
3 4 5 6 → length
7 → GAME
8 → ERROR
9 → NOT_STARTED(0x51)
10,11,12 → 0x62,0x6c,0x65