

Gra warcaby TCP

Opis protokołu komunikacyjnego

Aplikacja używa protokołu TCP by przesyłać ruchy graczy między klientami a serwerem, Są używane następujące typy komunikatów:

Wysyła klient do serwera

Do przesyłania ruchów na planszy między klientami a serwerem

IIDXYAB

Gdzie znaki kolejno oznaczają: I - id gracza, D - kolor pionków gracza, XY - współrzędne ruszanego pionka, AB - współrzędne docelowe.

Wysyła serwer do klienta

Do informowania klienta o niepoprawnie wykonanym ruchu

ERROR

Do akceptacji ruchu klienta

ACCEPT

Opis implementacji

Serwer

Użyta technologia: język C++

- Uruchomienie serwera odbywa się poprzez uruchomienie funkcji `main()`, która uruchamiając się rozpoczyna nasłuchiwanie. Połączenia są odbierane poprzez funkcję `accept()` i przekazywane do funkcji `handleConnection()`.
- Następnie nowe połączenie zaczyna być obsługiwane przez ww. Funkcję. Tworzona jest struktura `thread_data`, która zawiera informacje takie jak tymczasowy bufor, id gracza, id oponenta, czy deskryptor gniazda połączeniowego. Dalej ruch tworzonego wątku przekazywany jest do funkcji `ThreadBehaviour()`.
- Następuje główne dzianie wątku. Po inicjalizacji wszystkich potrzebnych struktur danych wątek wchodzi w główną pętlę programu
 - Na początku wątek zaczyna szukać swojego partnera do gry. Jeśli jest sam, zatrzymuje się oczekując na niego, gdy jednak już ktoś do gry dołączył dane wątki parują się - ustawiają swoje zmienne `oponent` jako deskryptor gniazda sparowanego przeciwnika. Pierwszy wątek który dołączył do gry ustawia pierwszą turę jako swoją.
 - Wątki sprawdzają czyja tura nastąpiła. Jeśli dany wątek wykryje turę przeciwnika, czeka aż przeciwnik dokona ruchu. Jeśli nie, wątek czeka na ruch, który zostanie wysłany przez przeglądarkowego klienta do niego. Gdy dostanie wiadomość następuje sprawdzenie jej poprawności. Następują dwie możliwości:
 - ACCEPT - stan planszy na serwerze jest aktualizowany i następuje tura przeciwnika

- ERROR - gracz musi wykonać ruch ponownie, tym razem poprawnie, tura się nie kończy, stan gry nie zostaje zaktualizowany
- Jeśli wszyscy gracze się rozłączą następuje koniec gry i serwer się wyłącza.

Klienci

Użyta technologia: język javascript wraz z Node.js

Uruchomienie klienta powoduje podpięcie się do portu serwera i wymusza na serwerze utworzenie wątku obsługującego właśnie utworzoną instancję klienta.

Po otwarciu okna przeglądarki na odpowiednim porcie użytkownikowi ukazują się przyciski/pola:

- From - lokalizacja ruszanego pionka,
- To - lokalizacja docelowa ruszanego pionka

Powyższe pola ustawiane są podczas klikania na odpowiednie pola.

- Submit - wysłanie danych do serwera (zebranych z pól From i To)
- Who am I? - informacja kim jestem Po kliknięciu aplikacja zaczyna pokazywać planszę do gry i informuje o kolorze pionków gracza.
- Check status - odpytanie o status ostatniego ruchu gracza
- Enemy move - aktualizuje ruch przeciwnika

Ponad planszą do gry pokazywana jest aktualna ilość pionków graczy. Zmieniają się one dynamicznie wraz z postępem rozgrywki.

Gdy jeden z graczy wygra grę, wyświetlony zostaje komunikat o wygranej. Gra się kończy.

Uruchomienie i kompilacja projektu

Kompilacja

Serwer kompilowany jest poprzez polecenie

```
g++ serwer.cpp -o serwer.out -pthread
```

Klienci w związku z użyciem języka javascript są tylko uruchamiani

Uruchomienie

Serwer:

```
./serwer.out 1111
```

Gdzie 1111 oznacza przykładowy port, na którym uruchamia się serwer

Klienci:

```
node ex.js 8000 1111
```

Gdzie 1111 oznacza przykładowy port na którym działa serwer, a 8000 oznacza przykładowy port, na którym działa klient.

Każdy klient potrzebuje inny port, np 8001, 8002...