

Temat zadania:

Gra planszowa Wybijanka

opis protokołu komunikacyjnego

Implementacja gry składa się z 3 komponent -

1. C++ serwer, który zawiera w całości business-logic gry, tworzenie "pokojów" itd. Serwer korzysta z wątków dla współbieżnej gry.
2. node-js serwer który łączy ostatnie komponenty. Komunikuję z C++ serwerem za pomocą unix-socket, z gui (stroną internetową) poprzez WebSockets.
3. GUI - strona internetowa - html + css + js.

Komunikacja odbywa się pomiędzy GUI a C++ serwerem, node-js serwer występuje wyłącznie jako adapter socket-WebSocket. Format wiadomości pomiędzy serwerem i Klientem nie znormalizowany. Jest to ciąg 100 bajtów, który jest przetwarzany do ciągu znaków.

Ponieważ to jest strategia „krok po kroku” z dwoma graczami, po każdym kroku serwer musi wysłać nowy stan gry użytkownikom. Początkowy stan planszy:

f-w-w-w-w

w-w-w-w-

-w-w-w-w

-----

-----

b-b-b-b-

-b-b-b-b

b-b-b-b-

Prefix `f` stosuje się do parsingu na stronie Klienta. `w` oznacza biały pion `b` - czarny, `-` puste pole.

Klient natomiast wysyła pożądany ruch w formacie `sourcedestination`. Na przykład `A2B3`. Serwer sprawdza czy taki ruch jest możliwy i jeżeli nie to wysyła do klienta wiadomość "Not a valid movement, please retry". Klient musi znów wysłać ruch. Jeżeli wysłany ruch jest poprawny, serwer zmienia stan planszy i wysyła graczom nowy stan. Oponent w tym momencie również dostanie wiadomość "your turn". Gra kończy się kiedy przynajmniej jeden z graczy nie ma ruchów. W tym przypadku wysyłane jest powiadomienie np. "won! white".

Po otwarciu strony internetowej, automatycznie odbywa się połączenie z serwerem C++ (przez nodejs serwer). Co drugi użytkownik uruchamia grę, w której będzie grał z poprzednim połączonym klientem.

Sposób instalacji oraz uruchomienia projektów umieszczony odpowiednio w folderach komponentów gry (README.md)