

Sprawozdanie z aplikacji klient – serwer „Kółko i krzyżyk”

Prowadzący: mgr inż. Michał Boroń

Autor: Kamil Luwański 136763

1. Opis protokołu komunikacyjnego

a) Komendy możliwe do wykonania przez serwer:

- „nx₁x₂” – rozpoczyna się nowa gra. Zaczyna x₁. Znak klienta to x₂
- „tx₁” – zmień turę na turę gracza x₁
- „sx₁i” – ustaw znak gracza x₁ na polu i, i=<0,8>
- „wy₁” – gracz y₁ wygrał
- „Sn” – szuka nowej gry

b) Komendy możliwe do wykonania przez klienta

- „e” – sam klient tego nie wysyła, ale taką wiadomość wypisuje sobie serwer, gdy klient się rozłączy
- „r(y|n)” – odpowiedź klienta, czy chce zagrać ponownie ry – tak rn – nie
- [0-8] – pole, na którym gracz, którego jest tura, chce zrobić ruch

c) Legenda

- x₁, x₂ – znaki graczy. Możliwe znaki to „x”, „o”, „n”(brak znaku)
- y₁ – „x”/„o” wygrał krzyżyk/kółko „X”/”O” analogicznie, tylko wygrana przez walkower (druga strona się rozłączyła”. „d” oznacza remis

2) Opis implementacji

a) Serwer

Cały serwer napisany jest C, stąd też wszystko napisane jest w jednym pliku. Id obiektów IPC przetrzymywane są w osobnym typie strukturalnym.

Część główna serwera działa na dwóch wątkach umieszczonych w funkcjach.

Pierwsza (matchClients) - wyciąga klientów z kolejki klientów oczekujących i dodaje ich do tworzonego wątku gry. Druga (acceptClients) – akceptuje połączenia od klientów i wysyła ich do kolejki klientów oczekujących na rozgrywkę.

b) Klient

Główna klasa wywołuje tylko klasę MainWindowFrame, która otwiera JPanel służący do połączenia się z serwerem. Po połączeniu (zainicjalizowaniu klasy ServerConnectioner) uruchamiany jest JPanel gry, który wyświetla aktualny stan gry w kółko i krzyżyk. Klasa AnimationDrawer dba o regularne rysowanie (przy zadanym FPS) aktualnego stanu gry. Klasa Game przetrzymuje stan obecnej rozgrywki i regularnie tworzy SwingWorkera MessageHandler, który odbiera informacje od serwera i je obsługuje. Dodatkowo klasa MainWindowFrame posiada dwa Listenery.

Jeden nasłuchujący kliknąć myszką i wysyłający do serwera zapytanie o możliwość wykonania ruchu. Drugi nasłuchujący, czy chcemy zamknąć MainWindowFrame. Jeśli tak, to zamyka on najpierw połączenie z serwerem.

Wszelka komunikacja z serwerem realizowana jest za pośrednictwem klasy ServerConnectioner.

3) Opis sposobu kompilacji i uruchomienia projektu.

a) Kompilacja

Serwer kompilujemy komendą „gcc -pthread -serwerwsp.c -o serwer.exe -Wall”.

Klienta kompilujemy poprzez otwarcie w środowisku NetBeans i kliknięcie przycisku build lub run (zielona strzałka na pasku narzędzi). W przypadku przycisku run projekt się odpali i otworzy się okno połączenia. Można je zamknąć – projekt jest skompilowany.

b) Uruchomienie projektu

Serwer odpalamy przy pomocy komendy „./serwer.exe”

Klienta odpalamy w środowisku NetBeans przy pomocy przycisku run, lub z terminala.

Musimy być w folderze Client/build/classes i wpisać komendę „java clienttictactoe.ClientTicTacToe.java”