Sprawozdanie z aplikacji klient – serwer „Kółko i krzyżyk”

Prowadzący: mgr inż. Michał Boroń

Autor: Kamil Luwański 136763

1. Opis protokołu komunikacyjnego

a) Komendy wysyłane przez serwer do klienta:

- „nx_1x_2” – rozpoczyna się nowa gra. Zaczyna x_1. Znak klienta to x_2
- „tx_1” – zmień turę na turę gracza x_1
- „sx_1i” – ustaw znak gracza x_1 na polu i, i=<0,8>
- „wy_1” – gracz y_1 wygrał
- „Sn” – szuka nowej gry
- „f” – ruch, którego chciał dokonać klient, był błędny

b) Komendy wysyłane przez klienta do serwera

- „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

- x_1, x_2 – znaki graczów. Możliwe znaki to „x”, „o”, „n”(brak znaku), np. tx_1 oznacza możliwości „tx”, „to” lub „tn”
- y_1 – „x”/„o” wygrał krzyżyk/kółko. „X”/”O” analogicznie, tylko wygrana przez walkower (druga stron 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 kliknięć 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 otworzenie 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.
W przypadku odpalenia klienta z terminala musimy być w folderze Client/build/classes i wpisać komendę „java clienttictactoe.ClientTicTacToe.java”