

Ogólny opis rozwiązania:

Program tworzy najpierw serwer (FinalServerX) który nasłuchuje i przyjmuje każdego agenta(FinalClient) który spróbuje się podłączyć (nie zaimplementowałem opcji dodawania nowych agentów przez obecnie podłączonych). Z każdym dodaniem nowego agenta serwer rozgrywa grę między nowym agentem a już obecnymi. Agenci losują swoje liczby. W ramach zapewnienia uczciwości serwer losuje liczbę (0-1) którą dodaje do zsumowanych liczb obu grających agentów. Wynik zostaje zapisany do pliku HTML. Po 10 sekundach proces klienta zostaje zakończony.

Szczegółowy opis rozwiązania:

Proces klienta (FinalClient) w momencie podłączenia się do serwera powoduje utworzenie wątku(ProcesKlientaX) który będzie obsługiwał agenta i umieszcza go w liście „gracze” w serwerze. Agenci podłączając się do serwera generują listę losowych numerów, po jednym na grę z każdym obecnym agentem na serwerze. Przy podłączeniu serwer wysyła polecenie „UPDATE” do każdego agenta. Agenci losują nową liczbę(„rzucana liczba”), która posłuży do zagrania z nowym agentem. Grę rozgrywa serwer, losując dodatkowo liczbę od 1 do 0(symulacja losowego wyboru osoby od której zaczyna się odliczanie) , co zapewnia brak możliwości oszustwa//przewidzenia wyniku. Po rozegraniu pojedynczej rozgrywki, wynik zapisywany jest do Stringa „wynikiDoHtml”, a po rozegraniu wszystkich gier, String jest zapisywany do pliku HTML. W pliku html gracze są opisywani za pomocą portów. Po 10 sekundach proces gracza spróbuje się zamknąć. Uda mu się to jeśli lista graczy z którymi gracz rozegrał grę („zNimiRozegrałem”) jest równa liście graczy na serwerze. Za każdym razem gdy gracz opuszcza serwer wywołuje statyczną metodę na serwerze „zamkniecie”. Metoda ta sprawdza czy opuszczający gracz jest ostatnim graczem na serwerze, jeśli tak, zostaje on(server) zamknięty.

Obserwacje, eksperymenty i wnioski:

Gracze odchodzą automatycznie po 10 sekundach, jeśli gracz byłby żywą osobą, sprawdzenie czy gracz napisał QUIT następowałoby w procesie ProcesKlientaX. Nie zaimplementowałem opcji JOIN, zgodnie z którą obecny gracz na prośbę nowego gracza mógłby go dodać do serwera. Jeśli gracz byłby żywą osobą dodałbym gui. Przy większej ilości graczy należałoby dodać kolejkę nowych graczy (tzn. dopóki pierwszy nowy gracz nie rozegra wszystkich gier, kolejny nowy gracz nie zostanie dołączony). W trakcie budowy programu zrobiłem proste gui, które „wydawało polecenia” wysyłając do serwera komendy (przypisane do guzików).