

## Cerință

Se vor crea două programe în Python 3.x care să permită un joc de tip „5 în rând” - adică „X și 0” modificat pentru o tablă de 5x5 în loc de 3x3, între 2 clienți (care pot fi pe mașini diferite în rețea), prin intermediul unui server.

## Descriere

Serverul va asculta pe cel puțin un port, și anume port= nr. echipei\*100 (pentru a fi >1024). Se pot folosi porturi suplimentare după dorință.

Echipele cu număr impar vor folosi protocolul UDP iar celelalte TCP – vezi funcțiile corespunzătoare în Lab 2. Echipele cu UDP nu trebuie să implementeze metode suplimentare de corecție a erorilor față de cei cu TCP.

Programul client va avea ca unic parametru adresa serverului (numerică sau DNS, de exemplu *client.py 141.85.131.150*). Programul server nu va avea parametri.

În total la server se vor conecta exact 2 clienți; primul client care se conectează va fi informat de către server că joacă cu „X” și al doilea cu „0”. Fiecare client va afișa tabla de joc și mutările efectuate pînă acum de către ambii jucători; serverul va afișa tabla de joc cu mutările celor 2 clienți. Afișarea se poate face în stil minimal, în mod text (se poate desena un caroiaj din caractere alfanumerice, dar este suficient dacă se desenează doar cele 5x5 poziții de joc, ocupate cu 0, X sau punct pt. căsuțele încă neocupate). La fiecare mutare, tabla va fi reafişată.

În cadrul clientului, modul de specificare de la tastatură a coordonatelor următoarei mutări este la alegerea voastră (și va fi documentat). Serverul va accepta de la tastatură o comandă pentru încheierea jocului.

Rolul serverului va fi să intermedieze jocul între cei 2 clienți:

- toate mutările de la clientul 1 la 2 și invers vor fi transferate prin intermediul serverului, întrucît clienții nu se conectează direct unul la celălalt, ci amîndoi la server
- serverul va accepta doar mutările valide de la clienți; în cazul în care un client încearcă să mute pe o poziție deja ocupată, va refuza mutarea și va aștepta în continuare o mutare validă de la același client; de asemenea, serverul nu va permite unui client să mute de 2 ori (peste rînd).
- după fiecare mutare primită, serverul va analiza dacă s-a ajuns în situația de cîștig sau de remiză, și în caz afirmativ, va trimite celor 2 clienți mesaje cu semnificația „ai cîștigat”, „ai pierdut” sau „remiză”; clienții vor afișa mesaje corespunzătoare pe ecran și jocul se va termina.
- în cazul în care serverul primește de la tastatura sa comanda de încheiere a jocului, el va transmite clienților cîte un mesaj corespunzător, iar aceștia vor acționa prin terminarea programului, după care serverul se va termina el însuși.
- serverul va afișa pe ecranul său cel puțin informații de debug referitoare la starea jocului (mesaje primite de la clienți, adresele acestora, etc).

Fiecare echipă trebuie să producă o pereche client-server compatibile între ele; întrucît sînt diferențe la numărul de port, dar vor exista și alte diferențe datorită faptului că nu este vorba de un protocol standard, nu se cere compatibilitate între clienții și serverele de la echipe diferite. Mai mult, voi verifica ca programele să nu fie identice între echipe!

Orice aspect necesar funcționării, care nu a fost specificat mai sus, este la alegerea voastră, și-l puteți comenta în fișierul sursă și/sau în *readme.txt*. În final trebuie produs un set de programe care să permită jocul efectiv în rețea.

## Mod de predare

În directorul personal al echipei curente (de pe *matrix*) se va găsi un director *temal* care va conține 3 fișiere:

- programul *client.py* (executabil)
- programul *server.py* (executabil)
- un fișier *readme.txt* care va avea două secțiuni:

1) descrierea pe scurt a modului de operare a programelor (ca și când fișierul meu cu cerințe nu ar exista; voi trebuie să concepeți programele pentru a putea fi folosite de utilizatori independenți, așa cum e cazul oricărui soft descărcat de pe Internet).

2) se vor documenta **mesajele de protocol** adică modul în care se transmit diferitele mesaje schimbate între client și server prin rețea: poziția X-Y la care se dorește mutarea, poziția X-Y la care a mutat oponentul, mesajele de eroare, câștig, pierdere sau remiză, orice alte mesaje necesare. Practic, aceste mesaje formează „protocolul X și 0”, care nefiind standard, trebuie documentat. Este încurajată existența comentariilor și în fișierul sursă.

## Mod de testare - notare

Pentru fiecare echipă:

- voi deschide 3 terminale (1 server, 2 clienți), voi intra în directorul personal al echipei
- voi citi instrucțiunile din *readme.txt*
- voi juca verificând că jocul decurge după regulile cunoscute

Punctaj: 25% existența fișierelor și calitatea documentației (inclusiv, și în special, documentarea mesajelor de protocol), 75% funcționarea.

Orice problemă care duce la nefuncționarea jocului va însemna pierderea tuturor punctelor pentru partea respectivă, întrucât aveți posibilitatea să testați voi-înșivă după aceste cerințe în contul vostru. Programele trebuie să funcționeze pe serverul *matrix* și să nu necesite instalări suplimentare sau modificări.

### **Indicații:**

Programele au nevoie să verifice minim 2 surse de posibil *input*: socketul de rețea (eventual mai multe socket-uri) și tastatura (*stdin*). Pentru aceasta, este recomandabilă folosirea metodei *select*. Aceasta permite în general „selectarea” uneia dintre mai multe surse de input, output sau eroare, în momentul în care acestea devin disponibile.

Un exemplu de folosire: <https://pymotw.com/3/select/>

Documentația completă: <https://docs.python.org/3/library/select.html?highlight=select#module-select>