**Rock Paper Scissors**

Aceasta documentatie descrie implementarea si utilizarea jocului „Rock Paper Scissors” sau „Piatra Hartie Foarfeca” cu doi jucatori, bazat pe retea. Include descrierea surselor de cod, explicatii despre structuri de date si comunicare, si instructiuni pentru rularea aplicatiei.

1. **Surse de cod**

Sursele de cod sunt impartite in urmatoarele fisiere:

* **server.py**: Codul serverului care gazduieste jocul si gestioneaza conexiunile dintre jucatori.
* **client.py**: Codul clientului pentru jucatori, incluzand interfata grafica.
* **game.py**: Clasa *Game* care gestioneaza starea jocului si logica de decizie.
* **network.py**: Codul care se ocupa de conexiunile dintre client si server.

1. **Documentatie**

**b1) Enunt problema**

* **Strategia de joc**: Jocul urmeaza regulile clasice ale „Piatra hartie foarfeca”. Jucatorii aleg una dintre cele trei optiuni si castiga conform regulilor traditionale (piatra bate foarfeca, foarfeca bate hartia, hartia bate piatra).
* **Numar de jucatori**: Jocul este conceput pentru doi jucatori conectati printr-o retea.
* **Castig**: Jucatorul care face alegerea castigatoare este declarat castigator. Daca ambii jucatori aleg acelasi lucru, jocul este un „tie” (egalitate).

**b2) Scurta descriere pentru proiect**

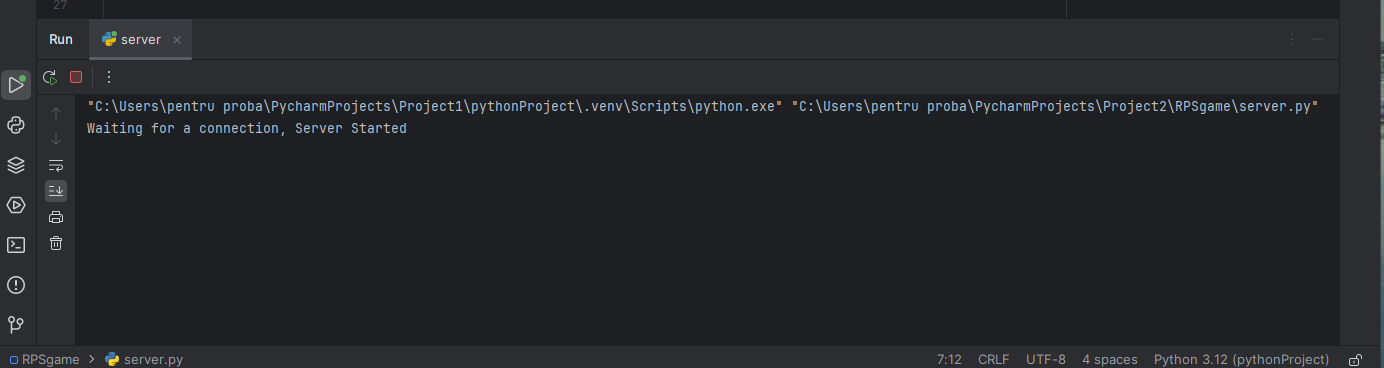
* **Implementare**: Serverul este construit folosind modulele **socket** si **\_thread** pentru a gestiona conexiunile si a permite jucatorilor sa se alature. Clientul este construit folosind **pygame** pentru interfata grafica. Serverul mentine o instanta a clasei *Game* pentru fiecare joc in desfasurare si permite jucatorilor sa interactioneze cu aceasta instanta.
* **Structuri de date folosite**: Clasa *Game* foloseste liste pentru a pastra alegerile jucatorilor, iar serverul foloseste un dictionar pentru a urmari jocurile in desfasurare. Comunicarea intre client si server se face prin **socket**-uri, cu date serializate si trimise cu ajutorul **pickle**.
* **Comunicare in retea**: Comunicarea se face prin conexiuni TCP/IP. Clientul se conecteaza la server, iar datele sunt transmise in ambele directii.

**b3) Tabel cu structuri de limbaj**

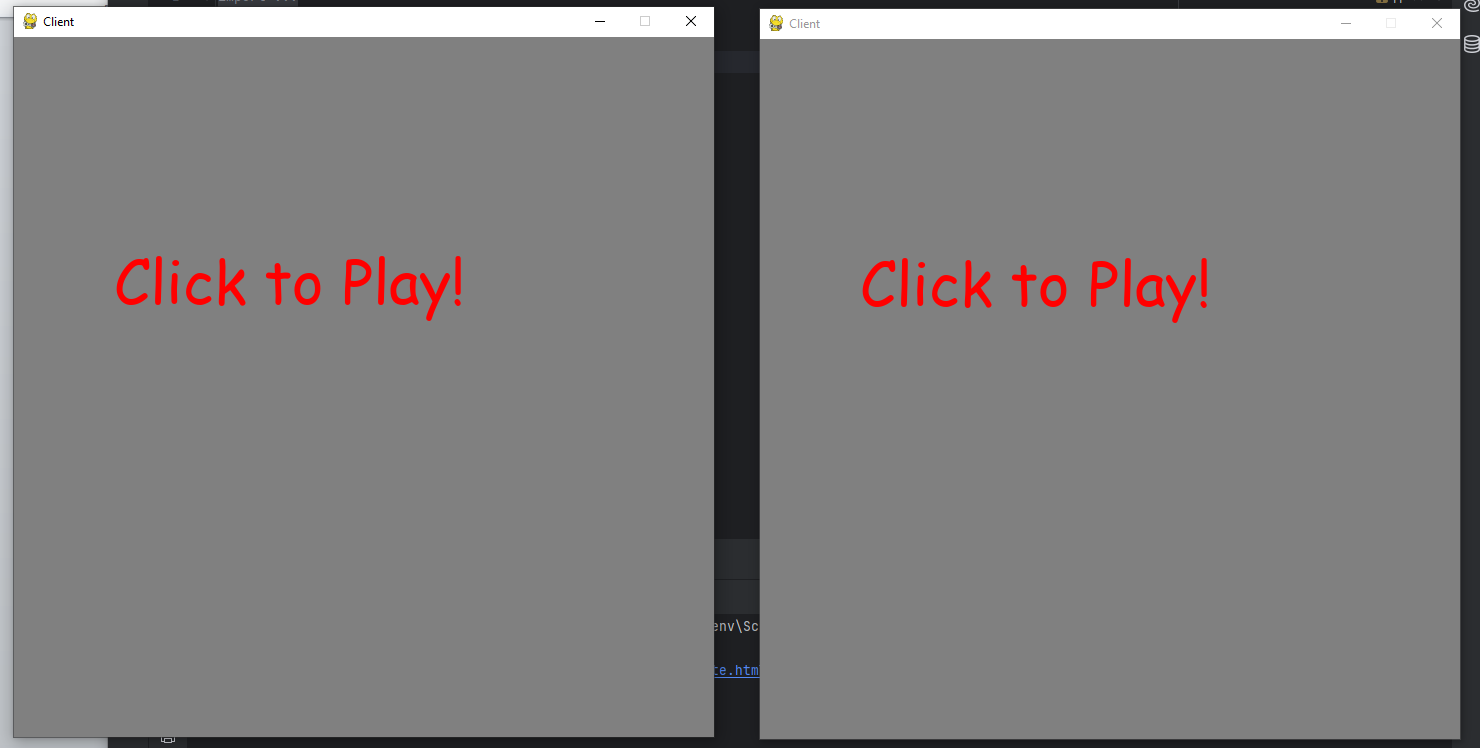
|  |  |  |
| --- | --- | --- |
| **Structura** | **Descriere** | **Exemple de utilizare** |
| Liste | O structura de date secventiala care poate contine mai multe elemente, de obicei de acelasi tip. Listele pot fi ordonate sau modificate. In acest proiect, ele stocheaza miscarile jucatorilor si rezultatele meciurilor. | - *self.moves = [None, None]* – Stocheaza miscarile ambilor jucatori.  - *self.wins = [0, 0]* – Numarul de victorii ale fiecarui jucator. |
| Dictionare | Colectii de perechi cheie-valoare, cu chei unice. In acest proiect, dictionarele sunt utilizate pentru a gestiona jocurile active pe server. | - *games = {}* – Dictionar pentru jocurile active.  - *games[gameId] = Game(gameId)* – Asociaza un ID de joc cu o instanta a clasei *Game*. |
| Socket-uri | Canale de comunicare intre doua dispozitive conectate prin retea. In acest proiect, socket-urile sunt folosite pentru comunicarea dintre client si server. | - *socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)* – Creeaza un socket TCP/IP.  *- self.client.connect(self.addr)* – Clientul se conecteaza la server. |
| Module externe | Module si biblioteci externe utilizate pentru functionalitate suplimentara, precum interfata grafica si manipularea datelor. | - *import pygame* – Folosit pentru interfata grafica a clientului.  - *import pickle* – Folosit pentru serializarea si deserializarea datelor transmise prin socket-uri. |
| Functii | Blocuri de cod care executa operatiuni specifice si pot returna valori. In acest proiect, functiile sunt utilizate pentru a organiza logica jocului si interactiunea client-server. | - *threaded\_client(conn, p, gameId)* – Gestionarea unui client intr-un fir de executie separat.  - *redrawWindow(win, game, p)* – Reincarca fereastra grafica in functie de starea jocului. |

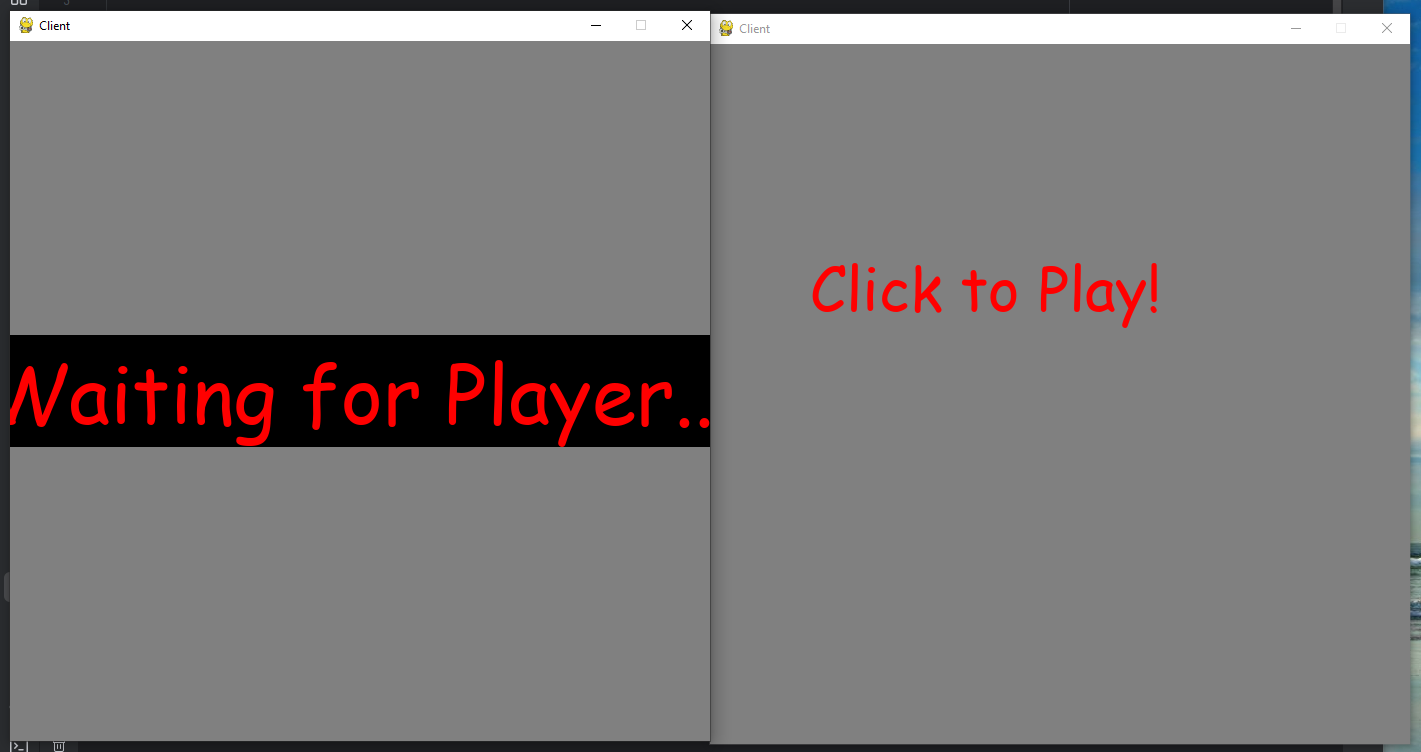
**b4) Capturi de ecran**

* **Stare server**: Afisarea mesajului „Waiting for a connection, Server Started”.

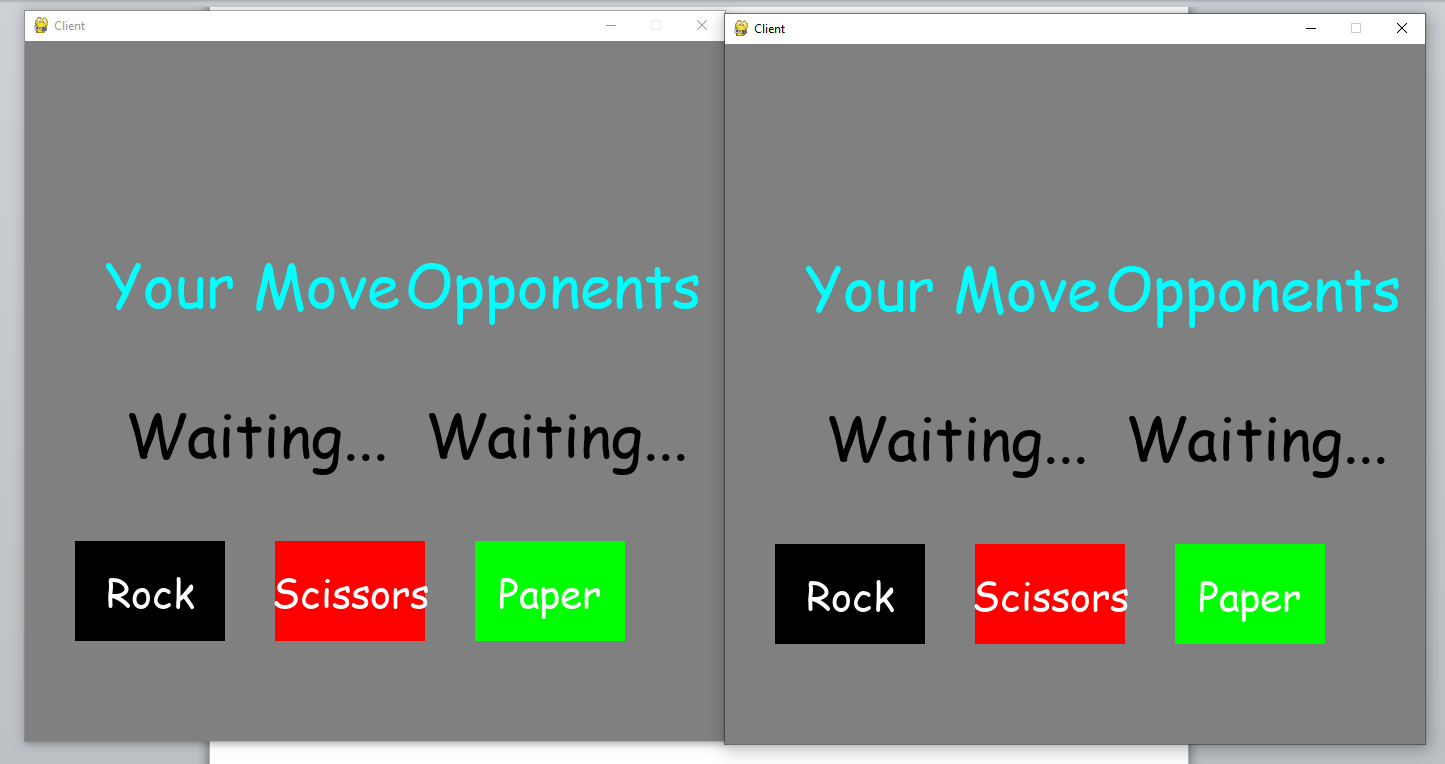


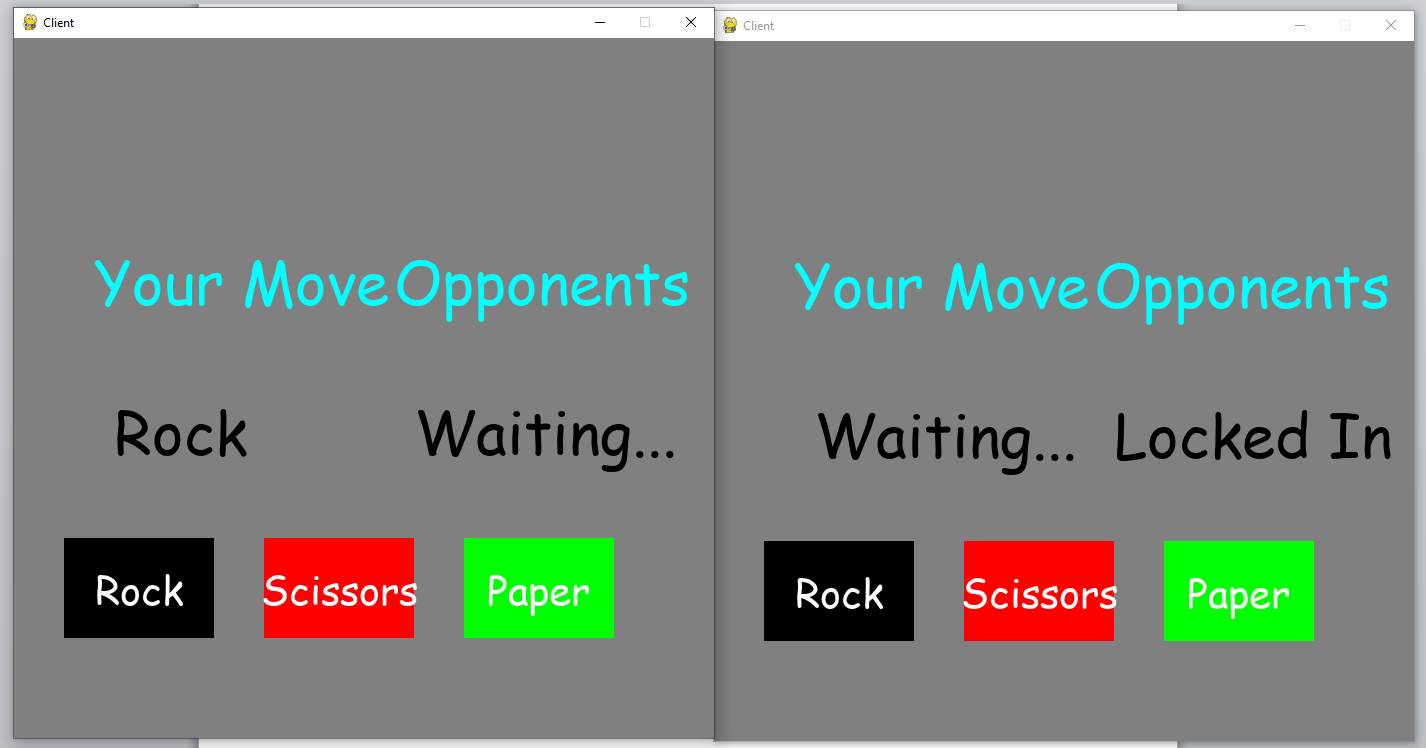
* **Conectare joc**: Fiecare jucator trebuie sa dea click petru a se conecta la retea si pentru a-si astepta adversarul.





* **Miscarea jucatorului**: Butoanele pentru alegerea pietrei, hartiei sau foarfecii.





* **Rezultat joc**: Afisarea mesajului „You Won!”, „Tie Game!”, sau „You Lost!”.

