

# ChessC Am incercat iar, tot ma lasa

Carp Marcel

December 2023

## 1 Introducere

Bla Bla

Ca alegere pentru tema 2, am decis să dezvolt ChessC, clientul dedicat serverului de șah ChessS. Clientul ChessC este proiectat să comunice eficient cu serverul ChessS, transmitând mișcările și primind actualizări ale stării jocului în timp real. Prin implementarea unei interfețe grafice atrăgătoare și intuitive, ChessC își propune să îmbunătățească experiența de joc și să faciliteze interacțiunea utilizatorilor.

## 2 Tehnologii utilizate

### 2.1 Server TCP concurent

ChessC utilizează protocolul TCP (Transmission Control Protocol) pentru a asigura o comunicare eficientă și fiabilă cu serverul ChessS. Alegerea TCP se datorează necesității unei comunicări ordonate și sigure, esențiale în transmiterea mișcărilor și primirea actualizărilor de joc. Beneficiind de comunicarea full-duplex oferită de TCP, ChessC permite o interacțiune bidirecțională și simultană între jucători și server, asigurând o experiență de joc fluidă. Jucătorii pot astfel efectua mutări, primi actualizări și comunica eficient fără întârzieri sau pierderi de date.

### 2.2 Socket-uri

ChessC se bazează pe utilizarea socket-urilor ca puncte finale esențiale pentru comunicarea în rețea între client și serverul ChessS. Aceste socket-uri servesc drept canale prin care se transmit mișcările de șah, actualizările stării jocului și alte informații critice pentru desfășurarea partidelor. Prin intermediul protocolului TCP/IP, socket-urile oferă o conexiune fiabilă și ordonată, fiecare participant la joc fiind identificat unic printr-o adresă IP și un număr de port.

## 2.3 CSFML

CSFML (Simple and Fast Multimedia Library pentru C) este o interfață în limbajul C pentru biblioteca SFML. SFML este o bibliotecă multimedia cross-platform dedicată dezvoltării de aplicații interactive și jocuri. Oferă capabilități pentru gestionarea ferestrelor, grafică 2D, audio și rețea.

## 3 Arhitectură aplicației

Aplicația ChessC funcționează reprezentând partea de client dintr-un model client-server, conectându-se la server pentru a participa la jocuri de șah. Clientul gestionează interfața utilizatorului și comunică cu serverul prin intermediul socket-urilor, asigurându-se că mutările și actualizările de joc sunt transmise și primite în mod eficient. Aceasta permite utilizatorilor să joace partide de șah în timp real, cu informații coerente și actualizate reflectând starea curentă a jocului.

### 3.1 Concepte implicate

- **Server**  
Serverul gestionează logica jocului și asigurându-se că toate mutările și interacțiunile sunt corecte și sincronizate. Clientul depinde de server pentru a participa în jocuri și pentru a comunica cu alți jucători.
- **Client**  
Clientul este interfața utilizatorului, prin care acesta interacționează direct cu jocul de șah. Acesta porneste și menține comunicarea cu serverul pentru a începe jocul, efectuează mutări și primește actualizări ale stării jocului. Este responsabil pentru prelucrarea și afișarea informațiilor de la server într-un mod intuitiv și utilizabil.
- **Adresa IP**  
Adresa IP reprezintă modalitatea prin care clientul se identifică și se conectează la server. Acesta utilizează adresa IP pentru a stabili o conexiune fiabilă și pentru a asigura că datele transmise și primite sunt direcționate corect.
- **Portul**  
Portul este o valoare numerică care identifică un punct final de comunicație specific pe server. În contextul TCP, fiecare adresă IP combinată cu un număr de port formează o adresă de socket unică, asigurându-se că datele transmise de client ajung la aplicația serverului corect. Clientul trebuie să fie configurat să utilizeze același port pe care serverul îl ascultă pentru conexiuni, asigurând astfel o comunicație eficientă.
- **Interfața**  
Aceasta este construită folosind biblioteca CSFML, care oferă o suită

de funcționalități pentru desenarea elementelor 2D, gestionarea evenimentelor de la utilizator și redarea sunetelor. Elementele grafice cum ar fi tabla de șah, piesele și meniurile sunt create și manipulate folosind ferestre, texturi și alte obiecte grafice oferite de CSFML. Interfața răspunde la acțiunile utilizatorului, cum ar fi clicurile mouse-ului sau apăsările de tastatură, și actualizează vizualizarea în consecință.

### 3.2 Diagrama aplicație

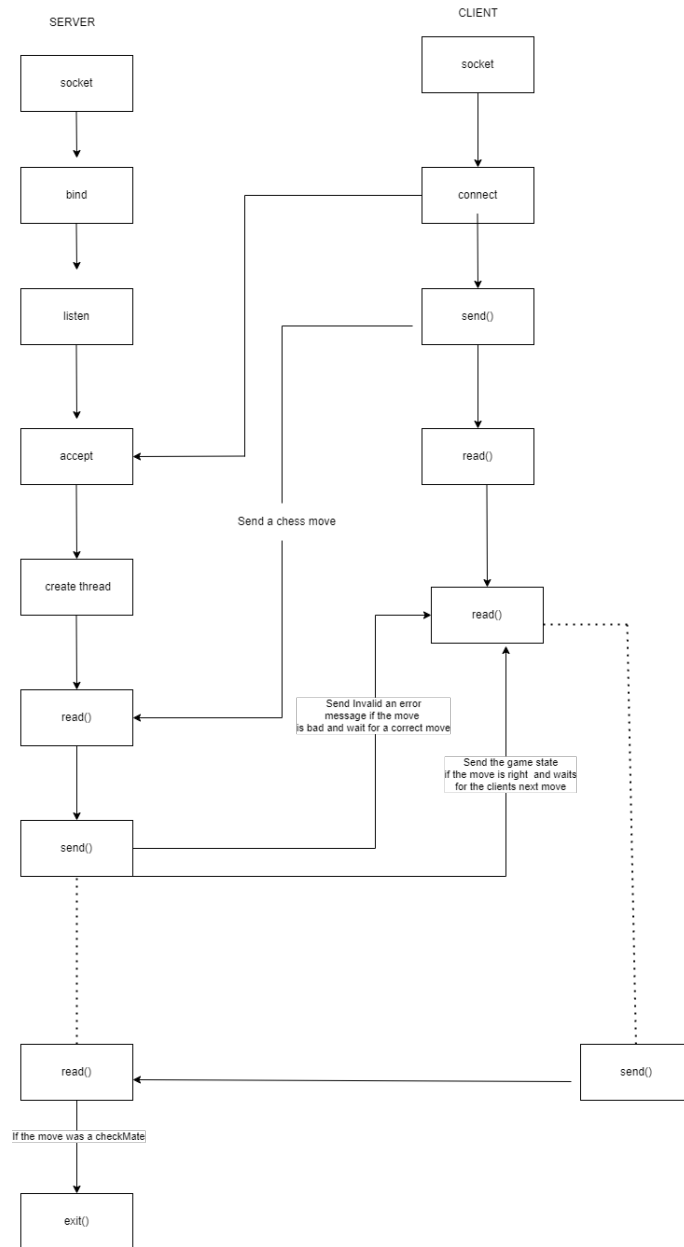


Figure 1:

## 4 Detalii de implementare

Comunicare dintre client și server se realizează prin socket-uri. Acest lucru avantajează scrierea programului deoarece nu mai trebuie să închidem capetele canalului de transmisie.

```
if ((sd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
{
    perror("Eroare la socket().\n");
    return errno;
}
```

Figure 2:

Funcția `connect` este apelată pentru a iniția o conexiune la server folosind socket-ul `sd` și informațiile adresei configurate anterior. Dacă conexiunea eșuează, programul afișează un mesaj de eroare și returnează codul de eroare asociat.

```
if (connect(sd, (struct sockaddr *)&server, sizeof(struct sockaddr)) == -1)
{
    perror("[client]Eroare la connect().\n");
    return errno;
}
```

Figure 3:

Pentru a controla desfășurarea jocului și a asigura o comunicare optimă între client și server, clientul `ChessC` utilizează diverse structuri de date, fiecare contribuind distinct la funcționalitatea și logica jocului de șah.

```
if (connect(sd, (struct sockaddr *)&server, sizeof(struct sockaddr)) == -1)
{
    perror("[client]Eroare la connect().\n");
    return errno;
}
```

Figure 4:

- **ChessPiece**

Structura `ChessPiece` reprezintă o piesă de șah individuală. Este definită prin două câmpuri: `type`, care stochează tipul piesei (rege, regină, turn, nebul, cal sau pion) și `color`, care indică culoarea piesei (alb sau negru).

- **Move**

Structura `Move` este folosită pentru a reprezenta și stoca informații despre o mutare în joc. Include coordonatele de început (`startX` și `startY`) și de sfârșit (`endX` și `endY`) ale mutării pe tabla de șah.

- **ChessBoard**

Structura ChessBoard reprezintă tabla de șah în sine, conținând o matrice bidimensională de ChessPiece pentru a urmări poziția tuturor pieselor. De asemenea, păstrează un întreg currentPlayer pentru a indica al cărui jucător este rândul să mute.

- **GameSession**

Structura GameSession gestionează o sesiune completă de joc între doi jucători. Include identificatori pentru fiecare jucător (player1 și player2) și o instanță a ChessBoard pentru a reține starea curentă a jocului. De asemenea, conține un câmp currentPlayer, care indică jucătorul care trebuie să mute în momentul actual.

```
24 ▾ typedef struct
25 {
26     char type;
27     int color;
28 } ChessPiece;
29
30 ▾ typedef struct
31 {
32     int startX;
33     int startY;
34     int endX;
35     int endY;
36     char promPiece;
37 } Move;
38
39 ▾ typedef struct
40 {
41     ChessPiece board[BOARD_SIZE][BOARD_SIZE];
42     int currentPlayer;
43 } ChessBoard;
44
45 ▾ typedef struct
46 {
47     int player1;
48     int player2;
49     ChessBoard game;
50     int currentPlayer;
51 } GameSession;
```

Figure 5:

## 4.1 Implementare functii

- **Functia pixelToChessCoords**

Funcția pixelToChessCoords transformă click-urile utilizatorului pe ecran în mutări de șah notate algebric. Calculează pozițiile de start și finish ale mutării pe baza locului unde utilizatorul a făcut click și pe baza dimensiunii căsuțelor de pe tabla de șah. Dacă click-urile sunt în afara tablei, returnează NULL. În caz contrar, generează notația algebrică a mutării

(de exemplu, "e2e4") și returnează acest șir de caractere pentru a fi folosit de restul programului.

- **Funcția deserializeGameSession**

Funcția `deserializeGameSession` este folosită în clientul `ChessC` pentru a transforma datele primite de la server, sub formă de text, înapoi în structura de date `GameSession`, care este folosită de client pentru a gestiona starea jocului.

- **Funcția drawChessBoard**

Funcția `drawChessBoard` este responsabilă pentru desenarea grafică a tablei de șah și a pieselor în fereastra de joc a clientului `ChessC`. Funcția inițializează formele și încarcă texturile necesare pentru a reprezenta piesele de șah (regina albă, regele alb, etc.), folosind imagini predefinite. Apoi pentru fiecare poziție de pe tablă, funcția determină ce piesă de șah se află la acele coordonate în structura `GameSession`. Apoi selectează sprite-ul corespunzător culorii și tipului piesei. La final, afișează toate schimbările în fereastra de joc.

- **Funcția main** Funcția `main` în clientul `ChessC` inițiază și gestionează întreaga sesiune de joc de șah, începând cu stabilirea conexiunii cu serverul și terminând cu gestionarea interfeței grafice și a logicii jocului. Inițial, funcția configurează conexiunea rețea prin specificarea adresei și portului serverului și apoi inițializează un socket pentru comunicare. După conectarea cu succes la server, clientul așteaptă să fie identificat ca jucător 1 sau jucător 2 și primește starea inițială a tablei de șah. Funcția ascultă răspunsurile serverului și actualizează starea jocului conform cu ultimele informații primite, re-desenând interfața grafică pentru a reflecta orice schimbări. Utilizatorul este capabil să interacționeze cu jocul prin intermediul unei interfețe grafice, care răspunde la clicurile și acțiunile acestuia, permițându-i să execute mutări când este rândul său. Funcția verifică în mod constant dacă este îndeplinită una din condițiile de terminare a jocului și închide conexiunea cu serverul și eliberează resursele utilizate odată ce jocul se încheie.

## 4.2 Scenarii de utilizare

### Conectarea la Server

- Utilizatorul deschide aplicația `ChessC` și este solicitat să introducă adresa și portul serverului. Odată introduse, clientul inițiază o conexiune la server și așteaptă confirmarea conectării. După stabilirea conexiunii, este gata să înceapă o sesiune de joc.

## Inițierea unei Partide de Șah

- După conectare, utilizatorul este atribuit ca jucător 1 sau jucător 2 de către server. Interfața grafică afișează tabla de șah inițială și utilizatorul așteaptă sau face prima mutare, în funcție de atribuirea ca jucător.

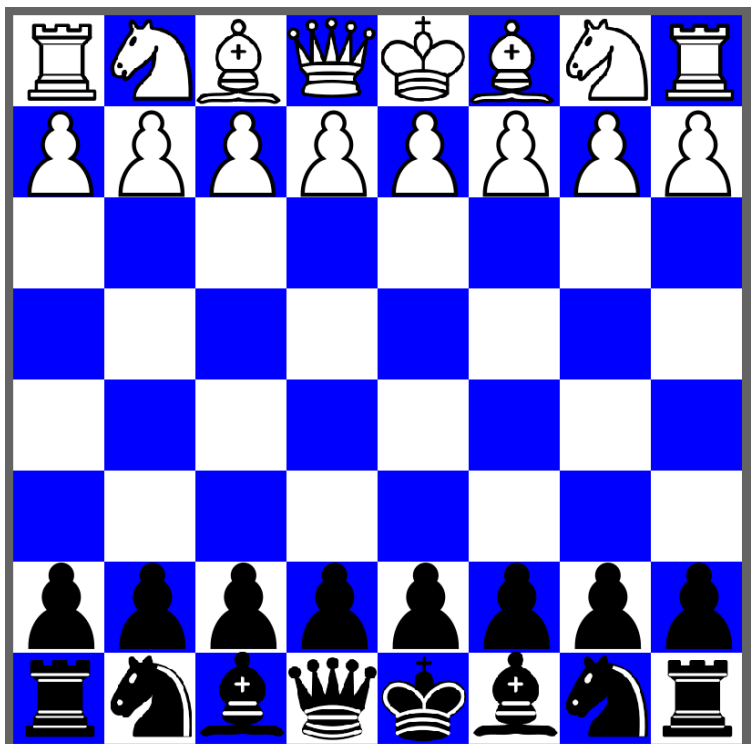


Figure 6:

## Desfășurarea Jocului

Utilizatorul efectuează mutări pe tabla de șah prin click-uri. Fiecare mutare este trimisă la server pentru validare și actualizarea stării jocului. Utilizatorul primește actualizări vizuale ale tablei de șah pe măsură ce jocul progresează, inclusiv mutările adversarului.



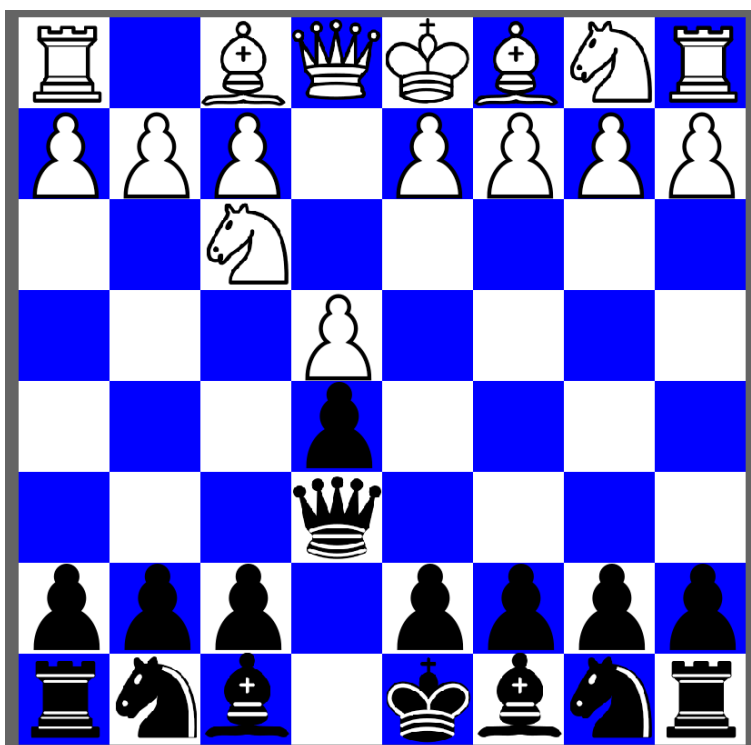


Figure 7:

## Terminarea Jocului

La finalul partidei, prin check-mate clientul închide conexiunea cu serverul și eliberează resursele utilizate. Utilizatorul este informat despre rezultatul jocului.



Figure 8:

## 5 Concluzii

ChessC reprezintă partea de client într-un mediu server-client pentru jocul de șah, exemplificând modul în care tehnologiile de rețea și interfața grafică utilizator se pot combina pentru a crea o experiență de joc captivantă și intuitivă. O funcționalitate ce ar putea îmbunătăți calitatea aplicației ar fi introducerea funcționalității de turneu dintre mai mulți clienți.

## 6 Bibliografie

1. <https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php>
2. [https://ro.wikipedia.org/wiki/Transmission\\_Control\\_Protocol](https://ro.wikipedia.org/wiki/Transmission_Control_Protocol)
3. <https://profs.info.uaic.ro/~eonica/rc/>
4. [https://ro.wikipedia.org/wiki/Port\\_\(re%C8%9Bea\)](https://ro.wikipedia.org/wiki/Port_(re%C8%9Bea))
5. <https://www.andreis.ro/teaching/computer-networks>