

Documentație Proiect DraughtsC

Grosu Victor-Alessandru
Grupa B4

Facultatea de Informatică, Iași

1 Introducere

Această documentație conține câteva informații despre proiectul "Draughts", motivația pentru care l-am ales, tehnologiile utilizate, concepte, detalii legate de implementare, scenarii de utilizare, modalități de îmbunătățire și bibliografia.

1.1 Motivația

Am ales acest proiect pentru a îmi demonstra capacitatea de a implementa o aplicație de tip joc în care utilizatorii pot interacționa într-un mod concurrent. De asemenea, proiectul este o oportunitate de a utiliza tehnologii ce țin de rețele de calculatoare și de a implementa un server care poate oferi mai multe servicii simultan.

1.2 Descrierea proiectului

1. Proiectul constă în interacțiunea dintre un server și mai mulți clienți. Fiecare client trebuie să își poată crea un cont, să se autentifice, să provoace alți jucători online la un joc sau să fie provocat, să joace jocul de Dame cu alți utilizatori și să consulte scorul tuturor jucătorilor.
2. În primă fază, utilizatorul este întâmpinat de o interfață grafică ce îi permite să introducă un nume de utilizator și o parolă sau să comute la modul înregistrare cont unde se adaugă încă un câmp pentru confirmarea parolei.

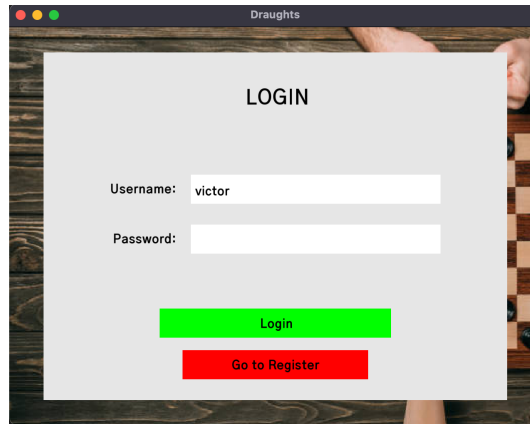


Fig. 1. meniul de logare

3. În urma logării, clientul își schimbă interfața într-o listă, unde utilizatorul poate vedea jucătorii online ce nu se află într-un meci. Acesta poate apăsa pe oricare din numele lor, pentru a îi provoca la un joc. Dacă clientul a fost provocat la un joc, acesta primește un mesaj care să aducă la cunoștință acest lucru utilizatorului, alături de opțiunile de a respinge sau a accepta jocul. Alternativ, în aceeași interfață, poate fi vizualizat și scorul tuturor jucătorilor.



Fig. 2. meniul principal

4. Dacă utilizatorul a acceptat un duel sau a provocat un adversar ce a acceptat duelul, interfața celor 2 clienți se modifică într-o tablă de joc, unde pot fi

mutate piese, se poate ieși din joc, se poate vedea câte piese sunt rămase sau ce jucător trebuie să facă o mutare.

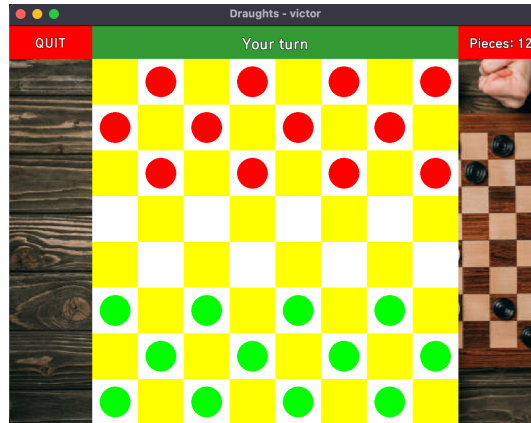


Fig. 3. desfășurarea jocului

2 Tehnologii utilizate

În cadrul proiectului "Draughts" am folosit următoarele tehnologii:

- C++ și SFML pentru implementarea aplicației client și a interfeței grafice.
- Protocolul TCP pentru comunicarea dintre clienți și server.
- Thread-uri pentru a permite serverului să gestioneze mai multe conexiuni simultan.

2.1 TCP

Protocolul TCP (Transmission Control Protocol) este un protocol de nivel de transport utilizat pentru transmiterea fidelă a datelor între sistemele informatice. Acesta oferă controlul fluxului de date și detectarea și corectarea erorilor. Aceste caracteristici sunt importante pentru aplicația noastră, deoarece vrem ca jocul de dame să fie transmis și recepționat corect între clienți și server.

2.2 SFML

Am ales să implementez aplicația client folosind librăria SFML (Simple and Fast Multimedia Library) pentru a crea interfața grafică. C este un limbaj de programare puternic, care îmi permite să implementez logica jocului de dame în mod eficient. SFML oferă instrumente pentru a crea rapid interfețe grafice intuitive și ușor de utilizat.

2.3 Thread

Am ales să creez câte un Thread pentru fiecare client ce se conectează la server, pentru a evita realizarea unui proces copil folosind fork(). Astfel, fiecare fir de execuție gestionează toată logica relevantă clientului de care se ocupă, fără a mai copia toate informațiile procesului părinte.

3 Arhitectura Aplicației

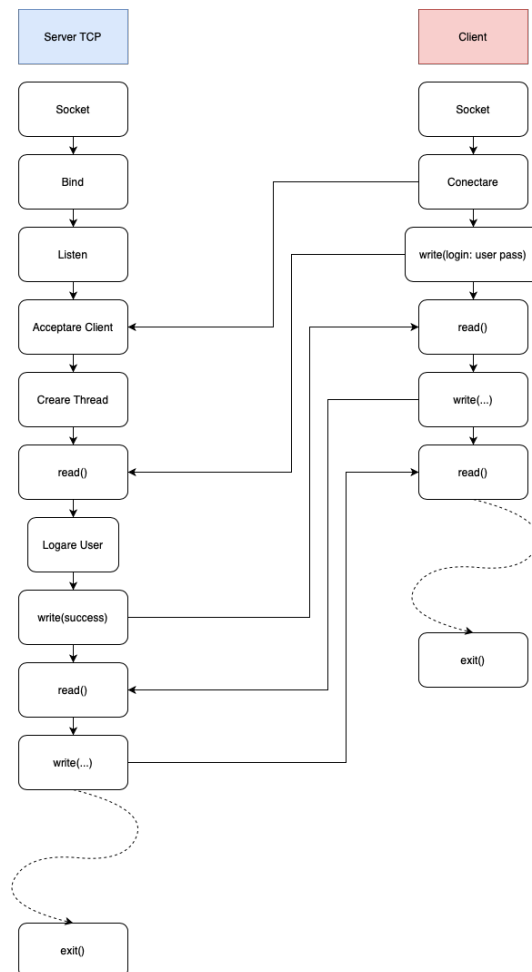


Fig. 4. logica comenzii "login"

Pentru comanda *"login: victor password"* după acceptarea clientului și crearea unui Thread, comanda este trimisă către server, care caută numele și parola introduse în lista utilizatorilor înregistrați. Dacă acesta există, serverul trimite un răspuns de tipul "success" către client și îl adaugă pe lista utilizatorilor conectați. În cazul în care utilizatorul nu a fost înregistrat anterior sau a introdus o parolă greșită, clientul va primi un mesaj de eroare, ce îi recomandă să se înregistreze sau să încerce o altă parolă.

4 Detalii de implementare

4.1 Scenarii de utilizare

- Clienții se conectează la server folosind IP-ul și portul necesar.
- Clienții trimit comenzi de tip "register: username parolă" pentru a-și crea un cont.
- Clienții trimit comenzi de tip "login: username parolă" pentru a-și adăuga numele pe tabelul jucătorilor online.
- Clienții pot trimite comenzi de tip "logout: username" pentru a se deconecta de la server.
- Clienții primesc informații despre ceilalți jucători conectați prin intermediul comenzii "getLobbyStatus" sau informații despre scorul tuturor jucătorilor înregistrați folosind "getLeaderboard".
- Clienții își caută un adversar din lista de jucători și lansează o invitație la joc trimițând o comandă de tip "challenge: username".
- Dacă clientul ales acceptă invitația, răspunzând cu "accept: username", clientul ce a provocat primește un mesaj de tip "accepted", iar cei 2 clienți dispar din lista de jucători ce pot fi provocați și încep un joc, primind de la server o matrice ce simbolizează tabla de joc. În cazul unui refuz, oricare dintre cei 2 jucători poate alege un alt adversar din tabelă.
- În cadrul unui joc, după fiecare mutare, clientul trimite matricea modificată către server prin intermediul comenzii "setMatrix: matrice", care la rândul său validează mutarea și trimite aceeași matrice celuilalt client.
- Dacă unul dintre clienți câștigă jocul, sau se obține o remiză, ambii jucători vor primi de la server, alături de matrice, și o informație de tipul "result: username1/username2/draw" sau "result: forfeit" în cazul în care unul dintre jucători părăsește jocul în timp ce acesta este în progres.
- După fiecare sesiune de joc, ambii jucători se pot întoarce la meniul principal folosind butonul QUIT ce apelează comanda "returnToLobby: username", care îi va pune înapoi pe lista jucătorilor ce pot fi provocați sau ce pot alege un alt adversar.

4.2 Reguli ale jocului

- Fiecare jucător își poate muta propriile piesele pe diagonală, un pătrat, dacă acesta este liber.

- Dacă pătratul este ocupat de o piesă inamică, dar următorul pătrat de pe aceeași diagonală este liber, jucătorul poate sări peste piesa inamică, astfel scoțând-o din joc.
- Dacă piesa unui jucător ajunge pe ultima linie, piesa devine un "rege", putând fi mutată pe diagonală în orice direcție, dar se respectă în continuare regulile menționate anterior legate de numărul de pătrate mutate.
- Un client este câștigător dacă elimină toate piesele adversarului, sau dacă adversarul nu mai poate face o mutare în cadrul rândului său, dar clientul încă are mutări posibile.
- Se consideră remiză dacă ambii jucători rămân cu o singură piesă sau dacă niciunul dintre jucători nu mai poate face nicio mutare.
- În cazul în care unul dintre jucători părăsește jocul, adversarul devine câștigător.
- Un câștig valorează 2 puncte, remiza 1 punct, iar înfrângerea este punctată cu 0, scorul putând fi consultat în secțiunea "Leaderboards" din meniul principal.

4.3 Cod

1. Partea de cod în care clientul trimite un mesaj și primește un răspuns.

```
int sendMessage(char sentMsg[]) {
    if (write (code.server.sd, sentMsg, MESSAGE_SIZE) <= 0) {
        perror ("[client]Eroare la sendto() spre server.\n");
        return errno;
    }
}

int sendReadMessage(char sentMsg[], char resMsg[]) {
    sendMessage (appendChars (sentMsg, "."));

    bzero(resMsg, MESSAGE_SIZE);
    if (read (code.server.sd, resMsg, MESSAGE_SIZE) < 0) {
        perror ("[client]Eroare la read() de la server.\n");
        return errno;
    }
    if (isBackendError(resMsg)) {
        setError(resMsg);
    }
}
```

2. Partea de cod în care utilizatorul apasă butonul de login/register, iar clientul trimite informația către server. În cazul în care jucătorul nu a completat toate câmpurile, a introdus informații ce nu se potrivesc, încearcă să se conecteze cu un user inexistent sau a utilizat o parolă greșită, va primi un mesaj de eroare aferent.

```
if (mouseIsOnItem(code.loginMenu.firstButton)) {
    onLoginFirstButtonClickHandler();
}
```

```

void onLoginFirstButtonClickHandler() {
    char res[MESSAGE_SIZE] = {'\0'};
    if(code.loginMenu.isLoginActive) {
        if(strlen(code.appProps.playerName) && strlen(code.
            loginMenu.password)) {
            sendReadMessage(appendChars("login: ", code.
                appProps.playerName, " ", code.loginMenu.
                password), res);
        } else {
            setError("Please fill in all the fields");
        }
    } else {
        if(strlen(code.appProps.playerName) && strlen(code.
            loginMenu.password) && strlen(code.loginMenu.
            confirmPassword)) {
            if(strcmp(code.loginMenu.password, code.loginMenu.
                confirmPassword) == 0) {
                sendReadMessage(appendChars("register: ",
                    code.appProps.playerName, " ", code.
                    loginMenu.password) ,res);
            } else {
                setError("Passwords do not match");
            }
        } else {
            setError("Please fill in all the fields");
        }
    }
    if(res[0] != '\0') {
        if(isBackendSuccess(res)) {
            code.appProps.appMenu = MAIN_MENU;
            window.setTitle(appendChars("Draughts - ", code.
                appProps.playerName));
        }
    }
}

```

5 Concluzii

5.1 Idei de îmbunătățire

- Implementarea unor funcții de statistică: O îmbunătățire posibilă ar fi implementarea unor funcții care să permită utilizatorilor să vadă statistici mai detaliate, cum ar fi numărul de jocuri câștigate, numărul de mutări efectuate și alte informații relevante. Aceste informații ar permite utilizatorilor să-și urmărească progresul în joc și să ia decizii mai bune în timpul partidelor.
- Implementarea unor funcții de socializare: În prezent, proiectul "Draughts" permite utilizatorilor doar să joace împotriva unor adversari aleși de ei. O

idee de îmbunătățire ar fi implementarea unor funcții care să permită utilizatorilor să comunice între ei în timpul jocului, să trimită mesaje de încurajare sau să discute strategii. Acest lucru ar spori experiența utilizatorilor.

Bibliografie

1. Pagina materiei Rețele de Calculatoare: <https://profs.info.uaic.ro/computer-networks/cursullaboratorul.php>
2. Documentația SFML: <https://www.sfml-dev.org/documentation/2.5.1/>
3. Protocol TCP: https://en.wikipedia.org/wiki/Transmission_Control_Protocol
4. Jocul de Dame: <https://en.wikipedia.org/wiki/Checkers>