**Documentație**

1. **Titlu Proiect: Chess game**
2. **Descriere:**

Acest proiect reprezintă o aplicație completă de șah cu interfață grafică, dezvoltată în limbajul C, ce permite jocul între doi jucători conectați în rețea (server și client). Interfața este realizată cu ajutorul bibliotecilor SDL2, SDL2\_image și SDL2\_ttf, iar comunicarea între jucători se face prin socket-uri TCP folosind Winsock2 (Windows).

Jocul oferă o experiență interactivă, cu meniu grafic, introducerea numelui jucătorului, validarea mutărilor conform regulilor oficiale de șah, afișarea grafică a tablei și pieselor, precum și salvarea rezultatelor fiecărei partide într-un fișier text.

Proiectul este destinat atât exersării programării orientate pe evenimente și grafice, cât și înțelegerii principiilor de bază ale comunicației în rețea și sincronizării între procese. Este ideal pentru uz educațional, laborator sau proiecte personale.

1. **Functionalitati:**
2. **Meniu grafic interactiv**: menu() – Afișează un ecran de start cu fundal, butoane "Play" și "Exit" și o casetă pentru introducerea numelui jucătorului.

A screenshot of a computer

AI-generated content may be incorrect.

1. Gestionează evenimentele de mouse și tastatură pentru selectarea butoanelor și introducerea textului.

A screenshot of a game

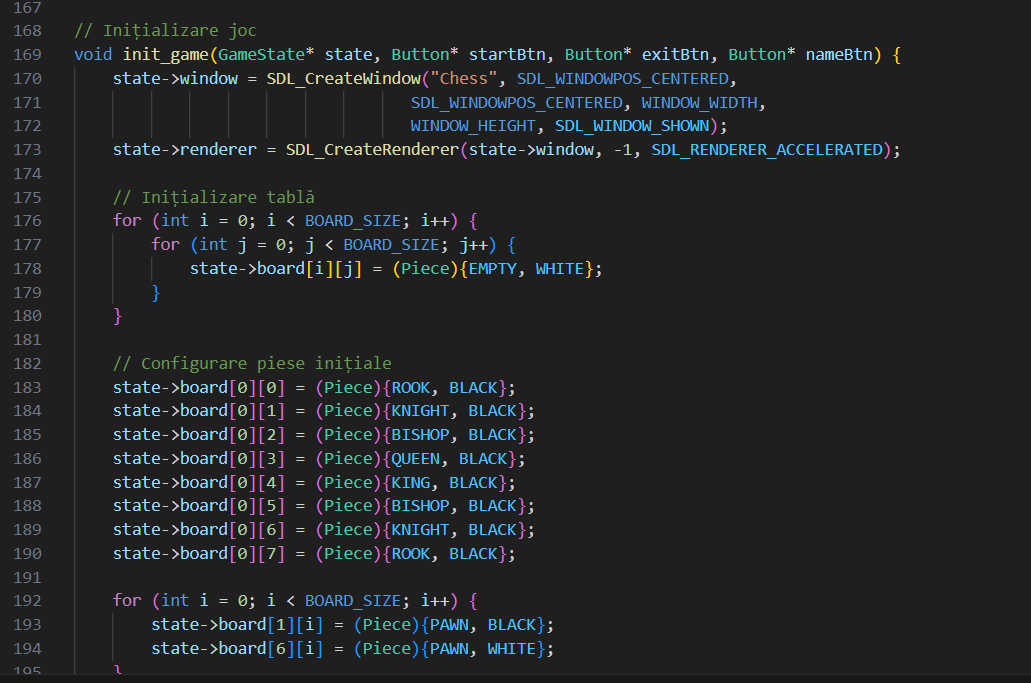
AI-generated content may be incorrect.

1. **A screenshot of a game

   AI-generated content may be incorrect.** La apăsarea "Play", numele introdus este trimis către server (dacă ești client) sau salvat local (dacă ești server).
2. **Inițializarea jocului:**

init\_game(GameState\* state, Button\* startBtn, Button\* exitBtn, Button\* nameBtn**)**

Creează fereastra și renderer-ul SDL, inițializează tabla de șah cu piesele în poziția de start, resetează variabilele de stare și configurează butoanele pentru meniu.



A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

1. Mutări și validare: User-ul selectează piesa pe care dorește să o mute, urmând ca selecția facută să fie interpretată.

A screen shot of a computer code

AI-generated content may be incorrect.

Fiecare tip de piesă de pe tabla de șah este tratată în particular, fiind implementate biblioteci. Structura bibliotecilor(corespunzatoare tipului de piesă) este asemanatoare: o funcție principală void move\_(tip\_piesa)() unde are loc mutarea piesei și o funcție care verifică dacă aceasta este validă, int is\_valid\_(tip\_piesa)\_move(), care returnează -1 dacă mutarea intenționată nu este posibilă.

* 1. PionA screen shot of a computer code

     AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

În funcție de culoarea piesei, direcția pionului poate fi y\_current + direction = 1 – negru, -1 -alb. Această funcție verifică pozițiile pe care pionul selectat poate ajunge și, totodată, dacă aceasta este și cea selectată de jucător.

* 1. Tură

A computer screen with colorful text

AI-generated content may be incorrect.

A computer screen shot of a computer code

AI-generated content may be incorrect.

* int is\_valid\_rook\_move(...):verifică ca “drumul” parcurs să fie liber și se asigură că în celula finală nu se afla o piesa de aceeasi culoare.
  1. A screen shot of a computer

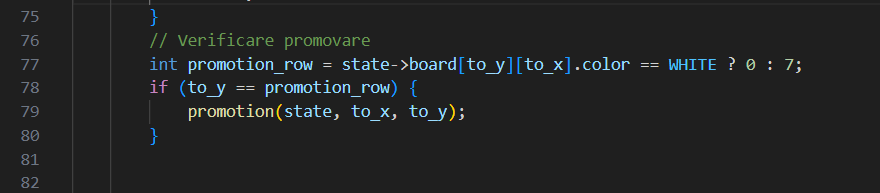
     AI-generated content may be incorrect.Cal
* int is\_valid\_knight\_move(...): generează toate pozițiile unde calul poate ajunge și verifică dacă se gasește cea dorită
  1. A screen shot of a computer code

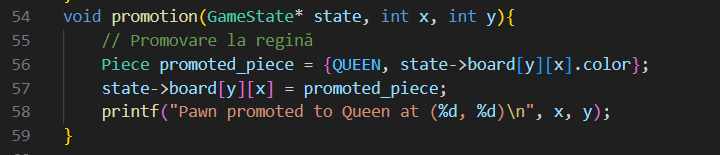
     AI-generated content may be incorrect.Nebun
* int is\_valid\_bishop\_move(...): verifică “drumul” – să fie liber.
  1. A screen shot of a computer code

     AI-generated content may be incorrect.Regina
* int is\_valid\_queen\_move(...):folosește funcțiile implementate pentru tură și nebun.
  1. A black screen with many colorful text

     AI-generated content may be incorrect.Regele
* Se generează pozițiile unde acesta poate să fie mutat, iar în cazul în care se gasește poziția dorita, returnează 1 – mutare validă.

1. Promovarea pionului

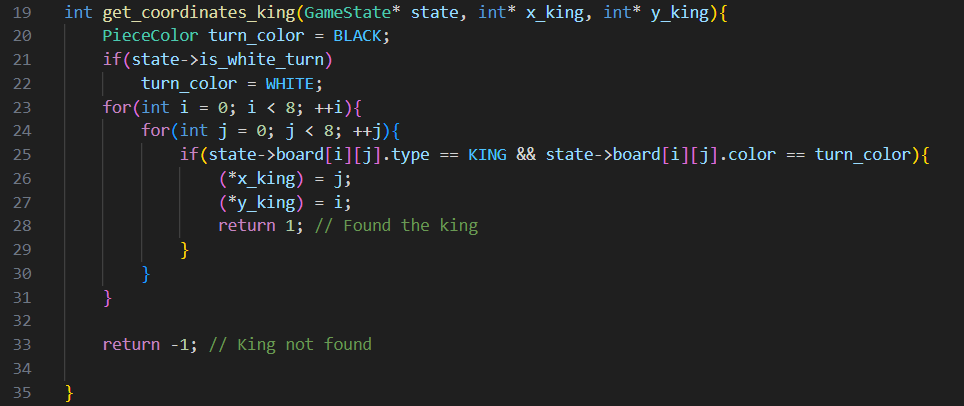




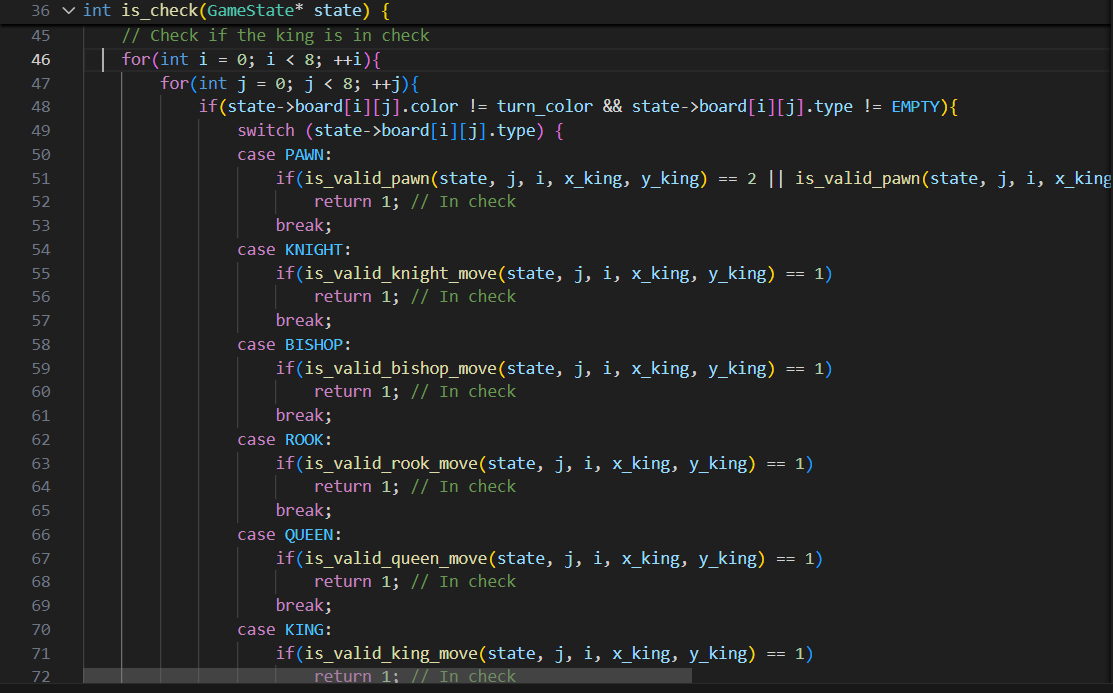
Promovează pionul ajuns pe ultima linie la regină (implicit).

Pentru o experiență completă, se poate implementa o funcție care permite user-ului să aleagă ce piesă să promoveze.

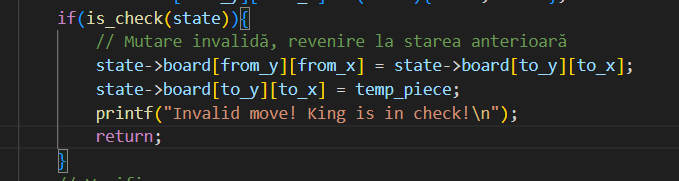
1. Detectare șah/șah mat



* int get\_coordinates\_king(...): caută pe toată tabla de șah poziția regelui jucătorului care trebuie să mute, pentru a verifica ulterior dacă acesta se află în șah



* parcurge toată tabla de șah și în funcție de piesa gasită(a adeversarului) verifică dacă poate ataca regele. În caz afirmativ se returnează 1 => regele este în șah. Această funcție este folosită și pentru a verifica dacă o piesă este blocată(pinned).



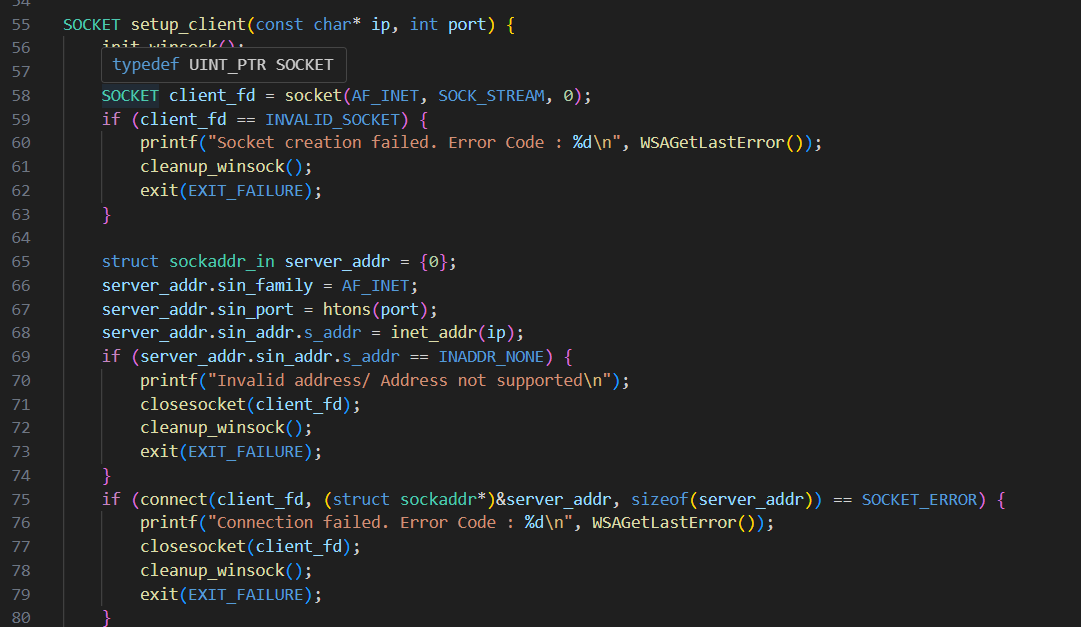


* int is\_checkmate(...): în această funcție se fac toate mutările posibile, iar dacă toate indică faptul că nu pot fi mutate => este șah mat, fiind câștigător adversarul. Spre exemplu dacă este albul la mutat, iar în urma generării tuturor mutărilor posibile nu poate găsi minim una validă, atunci înseamnă că negrul a câștigat.

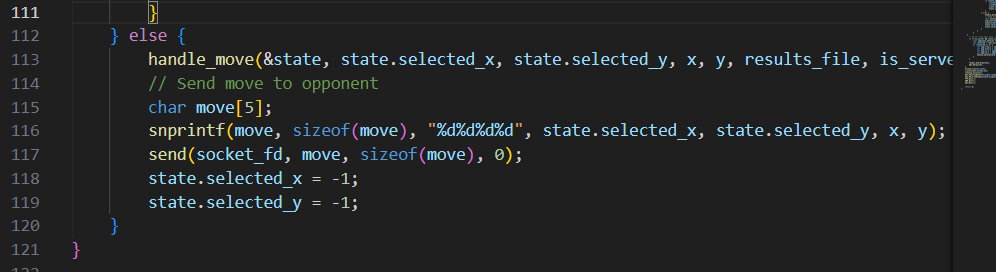
1. Multiplayer(socket programming):

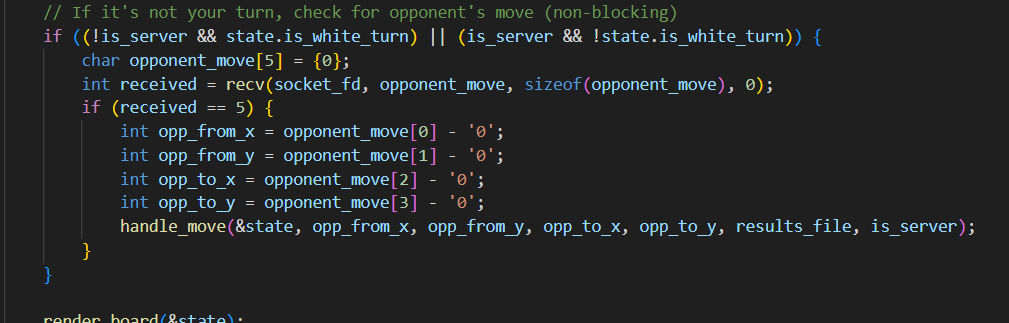
setup\_server(...), setup\_client(...):

Inițializează socket-ul pentru server/client și stabilește conexiunea TCP.



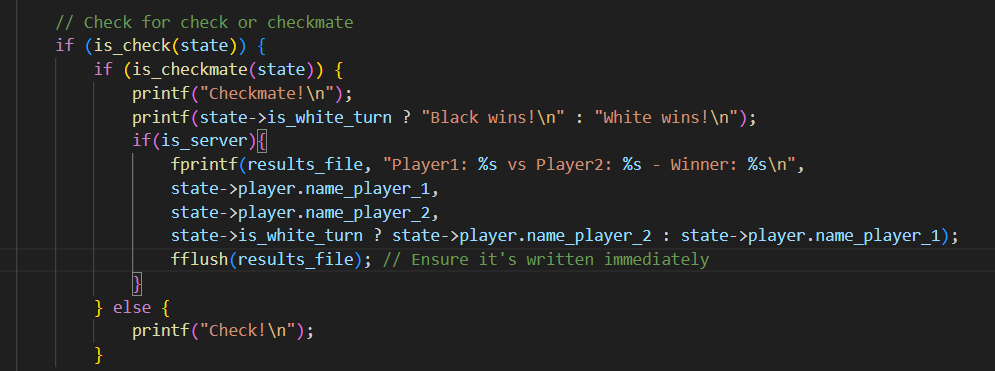
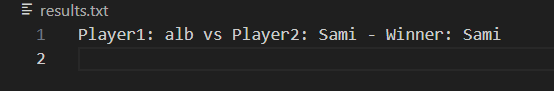


* send() și recv() trimit și primesc mutări și nume de jucători între client și server.



1. Salvarea rezultatelor

* Clientul îi trimite serverului numele introdus(clientul este întotdeauna jucătorul cu piesele negre). Când este șah mat, rezultatul jocului este salvat în fișierul “results.txt”.



1. **Instalare:**

1. Instalează dependențele:

- SDL2, SDL2\_image, SDL2\_ttf (prin MSYS2: `pacman -S mingw-w64-i686-SDL2 mingw-w64-i686-SDL2\_image mingw-w64-i686-SDL2\_ttf`)

2. Compilează proiectul:

- Folosește Makefile-ul:

**mingw32-make build**

3. Pornește serverul:

-Deschide un terminal în folderul unde se află jocul:

**./sdl\_game.exe server 12345**

4. Pornește clientul:

- În alt terminal sau pe alt calculator:

**./sdl\_game.exe client 127.0.0.1 12345**

- Înlocuiește `127.0.0.1` cu IP-ul serverului dacă nu rulezi local.

1. **Utilizare:**

* Meniu:
  1. Introdu numele în caseta de text.
  2. Click pe "Play" pentru a începe jocul sau "Exit" pentru a ieși.
* Joc:
  1. Selectează piesa cu mouse-ul, apoi dă click pe pătratul destinație.
  2. Mutările sunt transmise automat adversarului.
* Rezultate:
  + 1. La finalul fiecărei partide, rezultatul este salvat în `results.txt`.

1. **Structura Proiectului:**

Proiect\_chess/

│

├── assets/

│ ├── fonts/

│ │ └── arial.ttf

│ └── img/

│ ├── background/

│ │ └── menu\_background.jpg

│ └── pieces/

│ ├── white/

│ │ └── pawn.png, knight.png, ...

│ └── black/

│ └── pawn.png, knight.png, ...

│

├── src/

│ ├── main.c

│ ├── board.c

│ ├── board.h

│ ├── moves.c

│ ├── moves.h

│ ├── socket\_utils.c

│ ├── socket\_utils.h

│ ├── pawn.c, knight.c, bishop.c, rook.c, queen.c, king.c

│

├── results.txt

├── Makefile

└── README.md

1. **Extensii posibile:**
   1. Implementare AI pentru single-player.
   2. Adăugare sunete și animații.
   3. Salvare și încărcare automată a partidelor.
   4. Suport pentru chat între jucători.
   5. Diferite moduri de joc(blitz, rapid etc).
   6. Implementarea jocului pe timp.
   7. Posibilitatea ca user-ul să aleagă piesa dorită atunci când pionul ajunge în baza adversarului.
2. **Dependencies**

SDL2, SDL2\_image, SDL2\_ttf

Winsock2 (Windows, included with MinGW)

1. **Link Github:** [**https://github.com/samyro14/chess\_project**](https://github.com/samyro14/chess_project)

**Succes la șah!**