**Proiect Disciplina Tehnici de Programare**

**Joc de SAH in C**

Proiect realizat de Saftescu Tudor

In acest document voi detalia procesul de codare al jocului meu sah in C, care foloseste libraria Raylib si ofera utilizatorului posibilitatea sa joace sah atat in terminal, cat si intr-o fereastra cu o grafica minimalista, dar eleganta, avand posibilitatea sa joace atat singleplayer(contra unui AI cu 2 niveluri de dificultate), cat si multiplyaer, chiar daca inca nu este implementata o grafica la modul multiplayer.

Tehnologii utilizate in realizarea proiectului:

* C – limbaj principal
* Raylib – pentru interfața grafică (GUI)
* POSIX sockets (TCP/IP) – pentru multiplayer
* Linux/Windows – platformă de dezvoltare

Strctura proiectului intr-un mod simplist este:

Sah/

├── logica\_sah.c // Logica din spatele șahului

├── logica\_sah.h

│

├── sah\_terminal.c // Joc în terminal (local)

├── sah\_gui.c // Interfață grafică cu Raylib

├── sah\_ai.c //Logică AI

├── sah\_ai.h

├── sah\_multiplayer\_terminal.c // Multiplayer prin terminal cu sockets

│

├── assets/ //Imagini si fonturi pentru GUI

Dezvoltarea proiectului a avut loc pe parcursul semestrului al doilea si a constant in urmatorii pasi:

Etapa 1: sah\_terminal.c – Versiunea inițială

* Implementare completă a jocului de șah în terminal
* Control prin tastatură cu mutări în format "A2 A4"
* Reprezentare ASCII a tablei

Etapa 2: Separare în module

* Codul a fost modularizat:
  + logica\_sah.c și logica\_sah.h, care conțin toată logica jocului
  + sah\_terminal.c, adica interfața în terminal a fost păstrată minimalistă
* Avantaj: reutilizare a logicii în alte moduri de joc (GUI, AI, multiplayer)

Etapa 3: sah\_gui.c – Interfață grafică cu Raylib

* Tabla este desenată cu Raylib intr-o fereastra separata
* Utilizatorul mută piesele cu mouse-ul
* Sprijină toate regulile speciale
* Codul logic reutilizează direct logica\_sah.c

Etapa 4: sah\_ai.c și sah\_ai.h – Modul AI

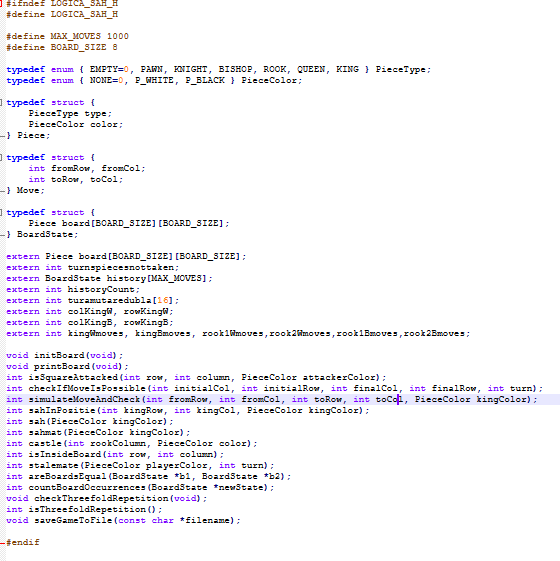
* Implementat un motor AI simplu:
  + Căutare pe mai multe niveluri
  + Evaluare statică de poziție
* AI-ul a fost integrat în sah\_gui.c ca mod PvC (Player vs Computer)

Etapa 5: sah\_multiplayer\_terminal.c – Modul multiplayer

* Refacere a interfeței terminal din sah\_terminal.c
* Adăugat suport pentru rețea (sockets TCP)
* Serverul joacă alb, clientul negru
* Mutările se transmit în format "A2 A4\n"
* Permite joc de la distanță prin rețea

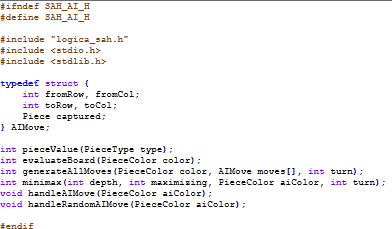
Acum voi prezenta cele mai importante si interesante bucati de cod din proiectul acesta, incepand cu cele 2 librarii facute special pentru proiect, anume logica\_sah.c si sah\_ai.c.

* 1. **„Logica\_sah.h ”** se prezinta asftel :



Această bibliotecă definește structurile și funcțiile esențiale pentru logica unui joc complet de șah. Include tipurile pentru piese, culori, mutări și stări ale tablei, alături de funcții pentru inițializarea jocului, afișarea tablei, validarea mutărilor (inclusiv rocada, șahul, matul și patul), verificarea repetării pozițiilor și salvarea partidei. Structura modulară permite separarea clară a logicii jocului de interfața grafică sau terminal. Această bibliotecă stă la baza tuturor versiunilor dezvoltate ulterior (terminal, GUI, AI, multiplayer), oferind un motor de joc robust și reutilizabil.

* 1. **„Sah\_AI.h”** arata asa:



Acest fisier definește baza functionalitatii pentru modul AI al jocului de șah. Conține funcții pentru evaluarea tablei, generarea tuturor mutărilor posibile și alegerea celei mai bune mutări folosind algoritmul Minimax. Structura AIMove reține detalii despre o mutare, inclusiv piesa capturată. AI-ul poate fi configurat să joace optim sau aleatoriu, în funcție de funcția apelată. Acest modul se bazează pe logica de bază din logica\_sah.h și este integrat în interfața grafică (sah\_gui.c), oferind o experiență completă de joc împotriva calculatorului.

* 1. **Raylib.h**

Cum aceasta nu este o librarie dezvoltata de acest proiect doar o voi descrie succint pentru a exprima ce poate face.

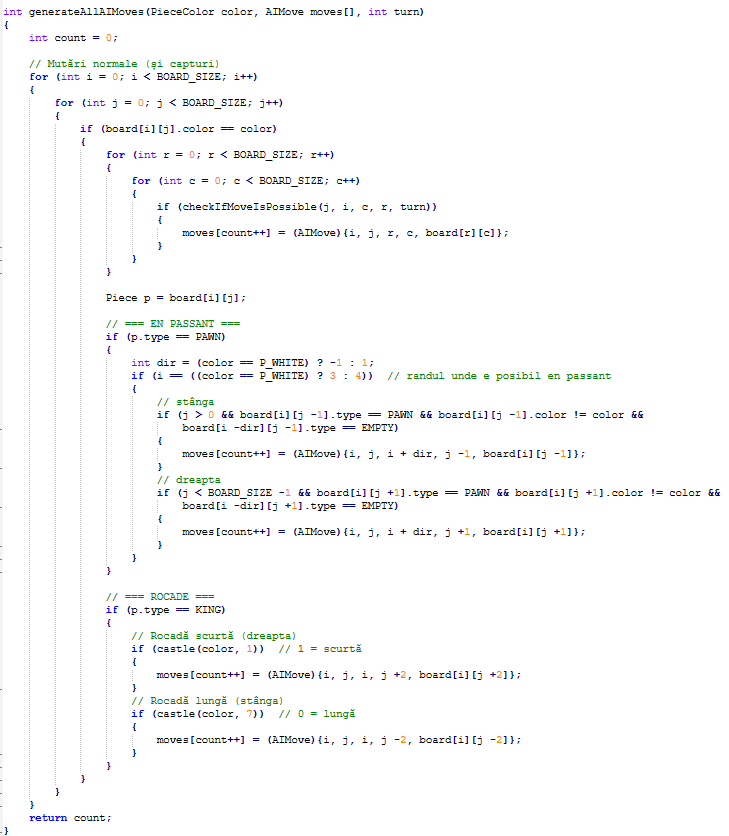
Este o bibliotecă C simplă și puternică pentru dezvoltarea de aplicații grafice și jocuri 2D/3D. Este ușor de folosit și ideală pentru începători, dar suficient de flexibilă si pentru proiecte avansate. Raylib oferă suport pentru desenarea de forme, texte, fonturi, sunete și interacțiuni cu tastatura sau mouse-ul. Poți crea jocuri de tip platformer, simulări, editoare grafice, interfețe animate sau chiar joculete cu interfață vizuală. Este multiplatformă (Windows, Linux, macOS) și vine cu documentație clară și exemple.

Acum trecem la fisierele „.c”, incepand cu

1. **„Sah\_ai.c”**

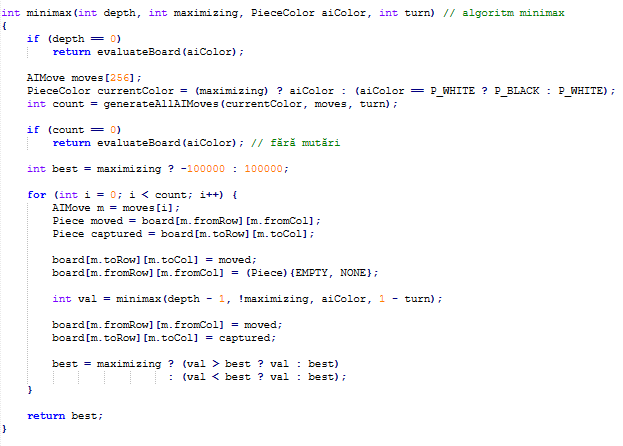
Functia in care cele mai importante 3 functii sunt

* generateAllAIMoves, care dupa cum genereaza numele creeaza un vector din toate mutarile posibile cu 4 for loop-uri, aceasta practic creează toate mutările posibile (inclusiv speciale: en passant, rocada).

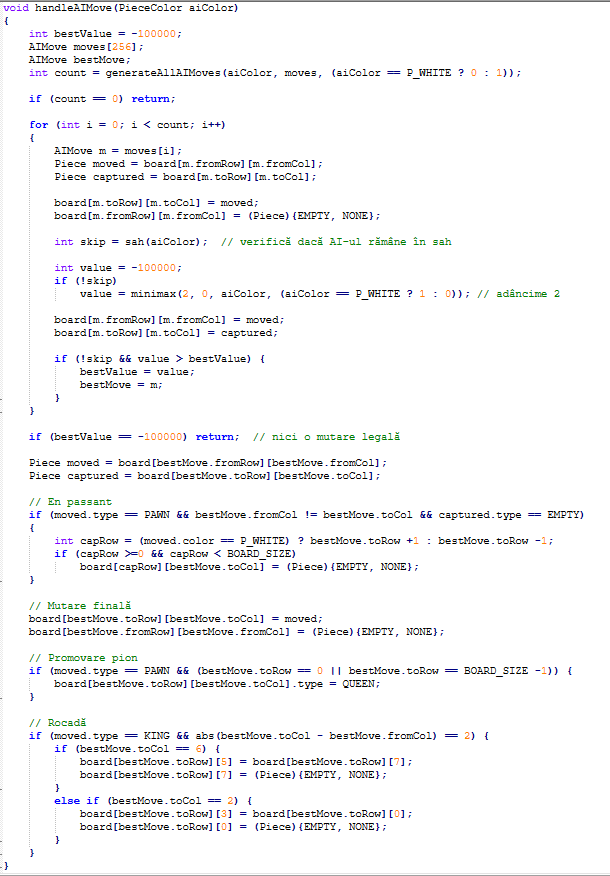


* Functia „minimax”:

Implementarea unui algoritm Minimax este esențial pentru deciziile AI-ului în jocul de șah. Scopule i este sa găseasca cea mai bună mutare simulând mai multe mutări în avans. Parametrii folositi de aceasta functie includ:

* depth: cât de departe "vede" AI-ul (adâncimea în arbore).
* maximizing: 1 dacă e rândul AI-ului, 0 dacă e al adversarului.
* aiColor: culoarea pieselor AI-ului.
* turn: cine are mutarea (0 = alb, 1 = negru).

Generează toate mutările posibile pentru jucătorul current, iar pentru fiecare mutare o aplică temporar, dupa care apelează recursiv minimax pentru mutarea adversarului si revine la starea anterioară. Dupa aceste apeluri, păstrează scorul minim/maxim (în funcție de jucător) si returnează valoarea cea mai bună găsită.

* Functia HandleAIMove si HandleRandomAIMove:

Funcția care execută cea mai bună mutare pentru AI, pe baza algoritmului minimax si generează toate mutările posibile pentru AI, iar pentru fiecare:

* Simulează mutarea.
* Verifică dacă AI-ul rămâne în șah (invalidează mutarea).
* Apelează minimax cu adâncime 2 pentru a evalua mutarea.
* Reține mutarea cu cel mai bun scor.

După ce a identificat-o, o aplică definitiv, fiind capabila sa gestionează si en passant, promovare, rocadă. Pe scurt e funcția centrală care ia deciziile inteligente ale AI-ului în timpul jocului.

1. **„Logica\_**sah.c”

Fisierul unde se intampla majoritatea functionalitatii jocului, chiar daca acest fisier nici macar nu este vazut de utilizatori. Cele mai interesante 3 functii din acest fisier sunt:

* checkIfMoveIsPossible:

Aceasta functie sta la baza oricarei mutari, ea verifică dacă o mutare este legală pentru piesa de pe poziția (fromRow, fromCol) către (toRow, toCol) în funcție de regulile șahului și de turul jucătorului (turn). Aceasta verifică toate regulile de mutare:

* tipul piesei (cal, pion, rege etc.),
* capturile legale,
* blocajele de pe traseu,
* mutările speciale (rocadă, en passant, promovare).  
  Este esențială pentru validarea mutărilor, atât în modul PvP cât și în AI.
* Sah/Sahmat:

Aceste 2 functii sunt necesare din cauza faptului ca fara sahmat nu putem stii daca meciul s-a incheiat, iar fara un sah functional nu putem stii daca nu cumva facem mutari „ilegale”. Verifică dacă jucătorul cu culoarea color este în șah(adică regele lui este atacat de o piesă adversă) sau daca si-a luat mat. Aceste 2 funcții sunt cruciale pentru:

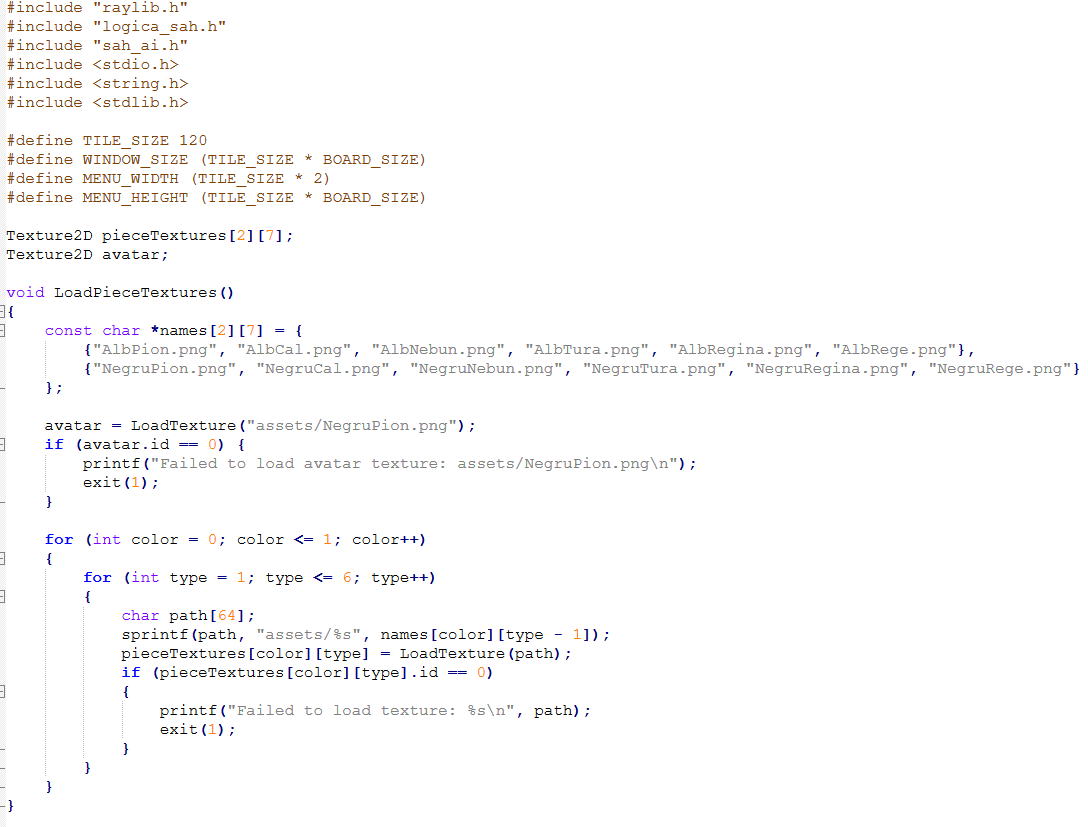
* a determina mutări legale (nu ai voie să lași regele în șah),
* a verifica condiții de șah mat sau remiză,
* a filtra mutările AI-ului (evită mutările care lasă regele în șah). Necesită o simulare internă a atacurilor posibile ale adversarului și localizarea regelui propriu.
* a finaliza in mod corect meciul.

Functia sahmat a fost probabil cea mai complicata functie din tot acest proiect din cauza numarului mare de cazuri speciale si de complexitate superioara, avand aproximativ 200 de linii de cod.

* Castle(rocada):

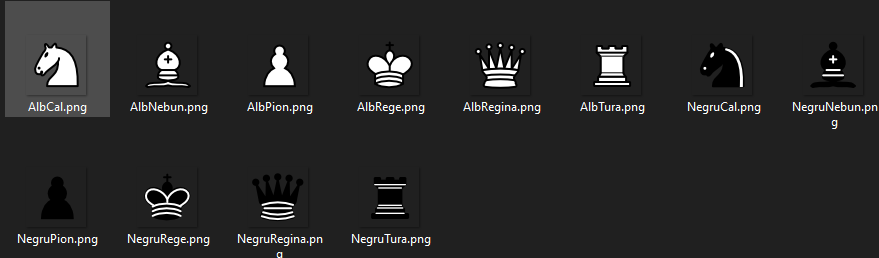
Verifică dacă jucătorul cu color poate face rocada de tipul tip (1 = scurtă, 0 = lungă). Această funcție gestionează o regulă rară, dar esențială în jocul real de șah și implică verificări multiple și subtile. Rocada este o regulă specială cu multe condiții:

* regele și turnul nu s-au mutat,
* nicio piesă între rege și turn,
* regele nu e în șah și nu trece prin șah.

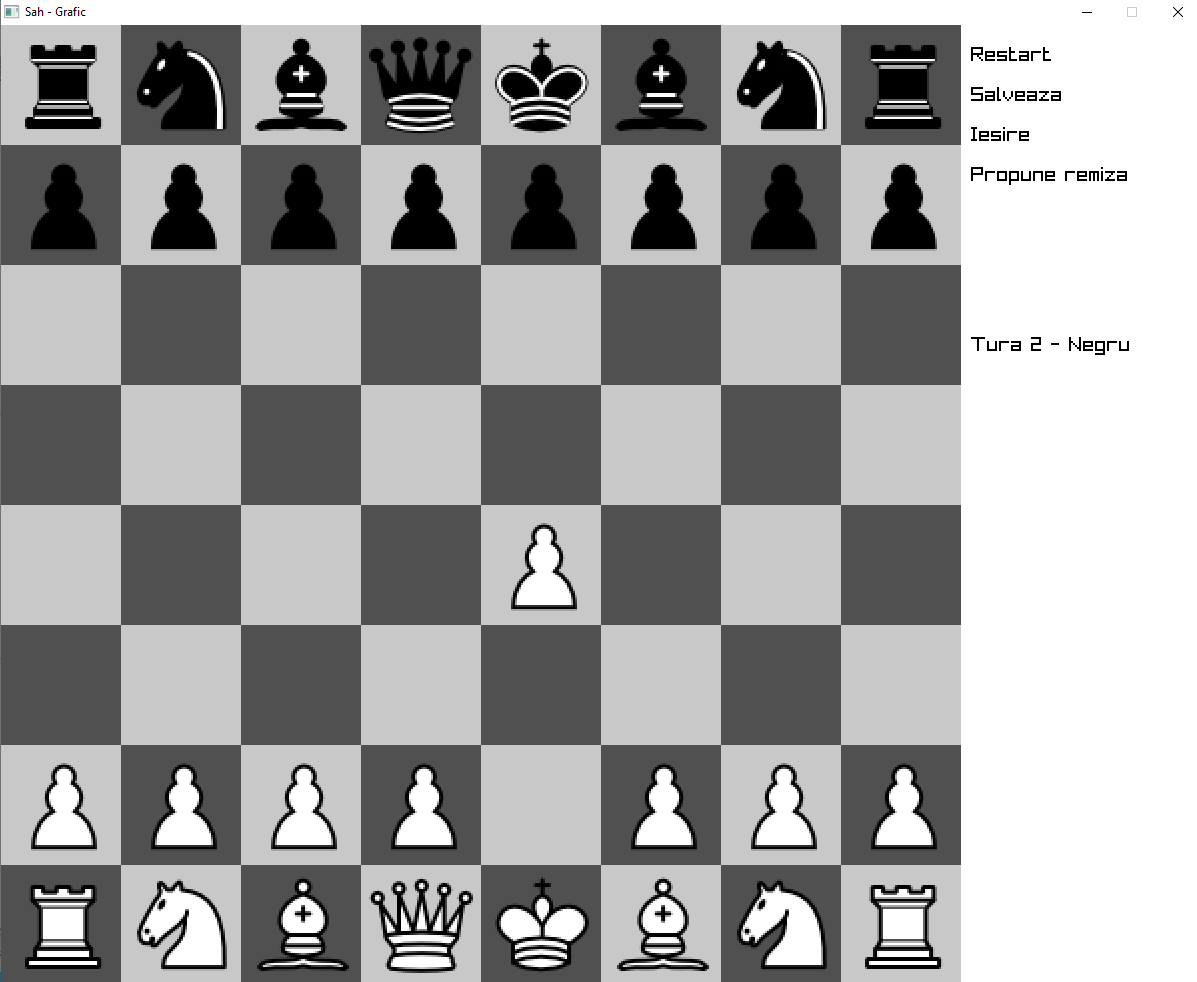


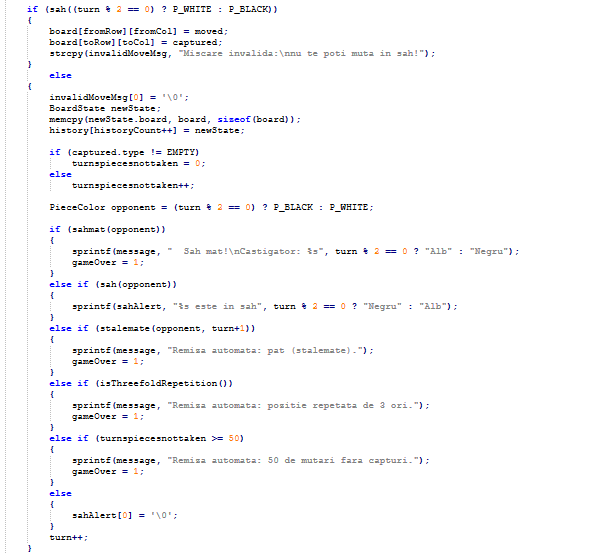
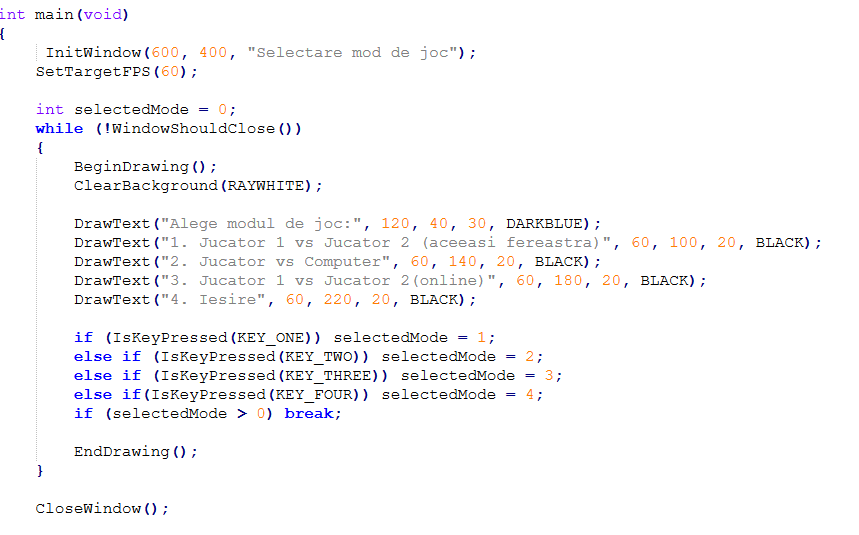
1. **"Sah\_GUI.c"**

Acesta este fisierul care face posibila folosirea sahului de catre utilizator. Acest fisier functioneaza astfel : intai cand se da « Run » ii da utilizatorului sansa sa aleaga ce doreste sa joace, avand de ales intre 1v1 pe aceeasi fereastra sau jucator vs calculator sau 1v1 pe 2 ferestre separate. Este cel mai lung fisier deoarece initializaeaza multiple ferestre si deoarece ruleaza intregul joc de pe ea.

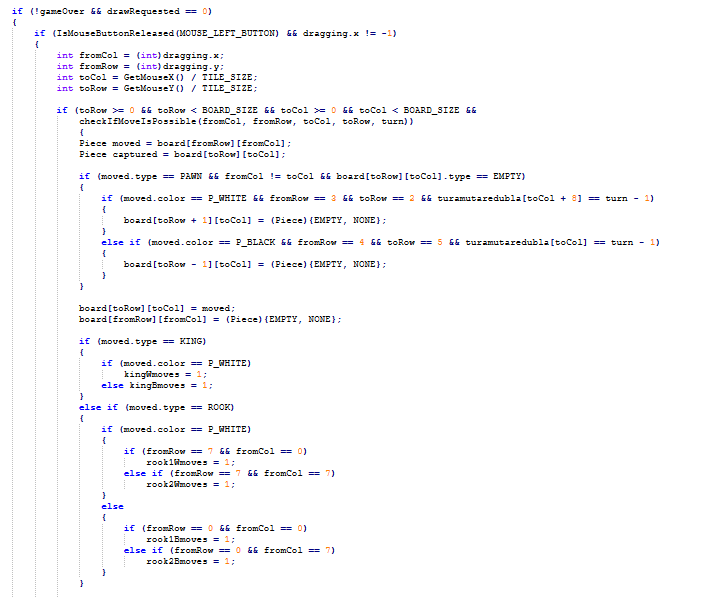
In Sah\_GUI.c, exista 2 functii facute pentru unload-ul si load-ul imaginilor de tip ong din folderul asociat numit “/assets” in care avem toate piesele.

Din punct de vedere al functionalitatii, in acest fisier folosim o multitudine din functiile Raylib pentru a avea o experienta placuta pentru utilizatori, deoarece asa putem muta piesele direct cu mouse-ul cu drag and drop. Mai jos este o imagine cu tabla si cum arata jocul:

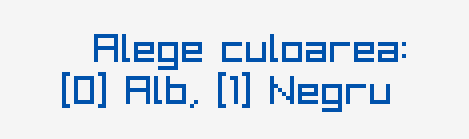


****Codul incepe cu alegerea modului de joc cu o fereastra auxiliara si o variabila numita „selectedMode” care are valori de la 1 la 4:

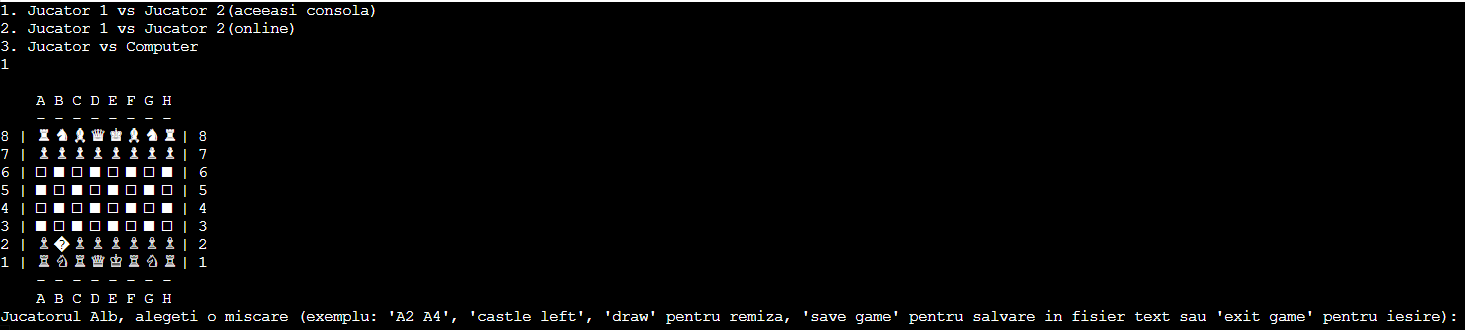
Interesant este si cazul in care unul din cei 2 jucatori este in sah, cand va fi afisat un mesaj in dreapta sus care anunta cine este in sah, moment in care functia sah previne realizarea mutarilor ilegale(anume cele care nu te scot din sah).



Asa este realizata mutarea cu drag and drop a pieselor, impreuna cu multe verificari pentru culoare.

Asemenea modului de joc din selectedMode==1, decurge si modul versus AI, care are 2 ferestre auxiliare, una pentru alegerea dificultati AI-ului (hard adica cel care face mereu miscarea optima si easy, adica cel care face mereu miscari random), dar si alegerea culorii cu care vrei sa joci, fiind posibil sa joci cu alb, dar si cu negru.

1. **"Sah\_terminal.c"**

Acesta este prototipul intregului joc, el fiind realizat astfel incat sa fie posibil un joc de sah in terminal, aspectul este rudimentar, dar codul din spate este extrem de complex, deoarece, asemanator cu Sah\_GUI.c se ocupa de intreaga functionalitate a tablei si a pieselor, regulile de joc fiind tot in Logica\_sah.c. Sah\_terminal.c a stat la baza tuturor celorlalte fisiere si este functional in continuare deoarece are inclusa libraria « logica\_sah.h ». Asa arata tabla si modul de introducere al mutarilor :

1. **"Sah\_multiplayer.c"**

Acest fisier reprezinta modul de joc multiplayer. Conexiunea Server-Client se realizeaza cu socket-uri. Iar ca arhitectura generală:

● Serverul (funcția run\_server\_game()) crează un socket, ascultă pe un port specificat și acceptă conexiuni de la clienți

● Clientul (funcția run\_client\_game()) se conectează la serverul folosind adresa IP specificată și portul prestabilit

Fluxul comunicării în multiplayer:

* Serverul se inițializează și așteaptă conectarea unui client
* Clientul se conectează la server
* Ambii jucători fac cate o mutare pe rand, incepand cu jucatorul alb(adica host-ul)
* Mutarile sunt transmise prin socket-uri
* Jucătorii fac cu rândul pentru a ghici cuvântul adversarului
* Tabla este modificata la fiecare pas si este transmisa între client și server

IP-ul 127.0.0.1:

● Acesta este adresa de loopback (localhost) care referă la calculatorul local

● Pentru server, codul folosește INADDR\_ANY (care permite acceptarea conexiunilor de pe orice interfață)

● Pentru client, utilizatorul introduce manual adresa IP a serverului

Sah\_multiplayer functioneaza doar pe Linux din pacate, dar voi realiza si varianta pe Windows cu grafici in Raylib pentru o experienta mai preietenoasa utilizatorilor.

**Concluzie :**

Dezvoltarea unui joc atat de complicat precum sahul intr-un limbaj precum C te poate invata rabdarea si stilul de lucru pentru proiecte mai amri, oferind o adevarata provocare, demna de un inginer. A fost o experienta placuta sa dezvolt acest joc de sah si consider ca inca il pot imbunatati pentru a fi si mai bun pentru utilizatori. In acest proiect am utilizat mult matrice si matematica booleana, dar si anumite tehnici de programare invatate la TP.