



Proiectare Software

Laboratory activity

Dezvoltarea unei inteligențe artificiale pentru jocul de șah

Nume: Trufin Radu-Sebastian

Grupa: 30233

Email: trufin.radu@yahoo.com



Enunțul problemei

Dezvoltați (analiză, proiectare, implementare) o aplicație **client** / **server** pentru un joc de șah. Implementarea se va realiza astfel încât un utilizator al aplicației (jocului) să joace cu calculatorul. Se va utiliza o variantă a algoritmului **MINIMAX**.

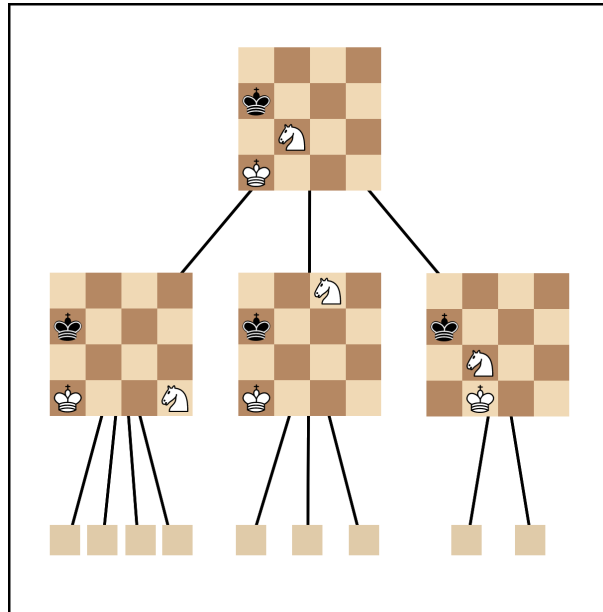


Figure 1: Algoritmul MINIMAX in sah

Aplicația va respecta șablonul Model-View-Controller și utilizează șablonul de proiectare creațional Singleton, șablonul de proiectare comportamental Observer și șablonul de proiectare structural Proxy.

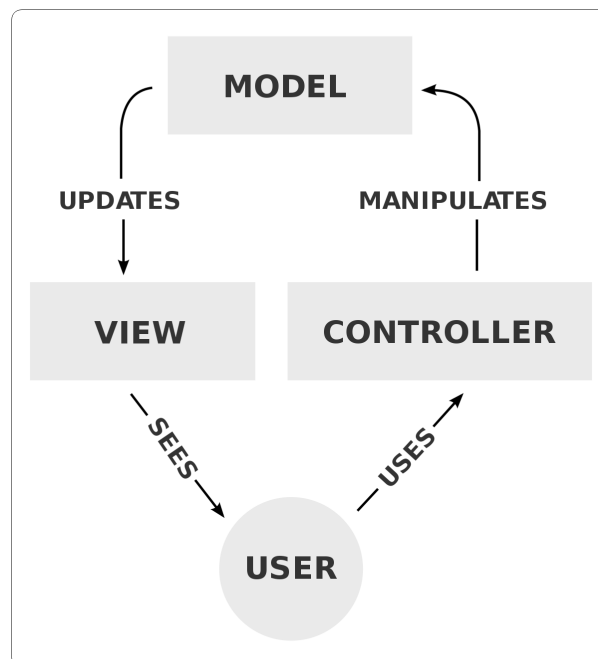


Figure 2: Sablonul Model-View-Controller

Analiză

În această fază se vor realiza diagramele cazurilor de utilizare specifice aplicației Client, dar și aplicației Server, după care se vor realiza diagramele de activități corespunzătoare tuturor cazurilor de utilizare. Pentru crearea diagramelor de clase, respectiv cazuri de utilizare (ulterior secvență și activități) specifice acestui proiect s-a ales aplicația **StarUML**. În cadrul diagramei cazurilor de utilizare corespunzătoare aplicației Client, entitățile care interacționează cu sistemul (actorii) beneficiază de următoarele:

1. Posibilitatea de a alege tipul oponentului (om sau calculator) și culoarea pieselor în cazul în care oponentul este un AI. Dacă utilizatorul dorește să joace împotriva unei inteligențe artificiale, acesta are posibilitatea de a alege dificultatea oponentului, prin modificarea adâncimii în cadrul algoritmului **MINIMAX**, adică numărul de apeluri recursive ale funcției. În mod implicit adâncimea este setată la 2, asta înseamnă că algoritmul **MINIMAX** va evalua și furniza calculatorului cea mai bună mutare considerând faptul că oponentul va juca optim următoarea mutare. Prin setarea unei adâncimi mai mari sau egale cu 3, oponentul va fi dificil de invins, însă timpul de execuție al programului va crește semnificativ, după cum era de așteptat. Acest caz de utilizare poate fi împărțit în 2 subcazuri (cel în care jucătorul uman alege să joace cu piesele albe și cel în care dorește să joace cu piesele negre). Acest caz de utilizare va fi numit "**Human vs AI**", cu subcazurile "**White vs AI**" și "**AI vs Black**".
2. Posibilitatea de a observa un joc între două inteligențe artificiale. În acest caz mutările vor fi realizate instant pentru o adâncime de 2. Pentru o adâncime de 3, un AI poate ajunge la o durată de până la 5 secunde pentru procesarea unei singure mutări. Acest caz de utilizare va fi numit "**AI vs AI**".
3. Posibilitatea de a exersa în mod liber (ultimul tip de joc este cel între doi oameni, în care fiecare jucător mută piesele pe rând prin intermediul mouseului, fără restricție de timp). Acest caz de utilizare va fi numit "**Practice**".
4. Posibilitatea de a încărca un joc în formatul PGN (Portable Game Notation) care este un format standard de text simplu pentru înregistrarea jocurilor de șah și este, de asemenea, acceptat de majoritatea motoarelor de șah. Indiferent de tipul de joc ales, după ce se termină un meci se generează un fișier text cu mutările în notație algebrică, respectând format. Acest caz de utilizare va fi numit "**Import PGN**".
5. Posibilitatea de a se abona la motorul de șah specific aplicației și a vizualiza starea poziției curente prin intermediul unei bare de evaluare (care afișează un scor pozitiv dacă piesele albe dețin un avantaj și un scor negativ dacă piesele negre dețin un avantaj). Acest caz de utilizare va fi numit "**Subscribe**".
6. Posibilitatea de a comunica cu alți clienți.



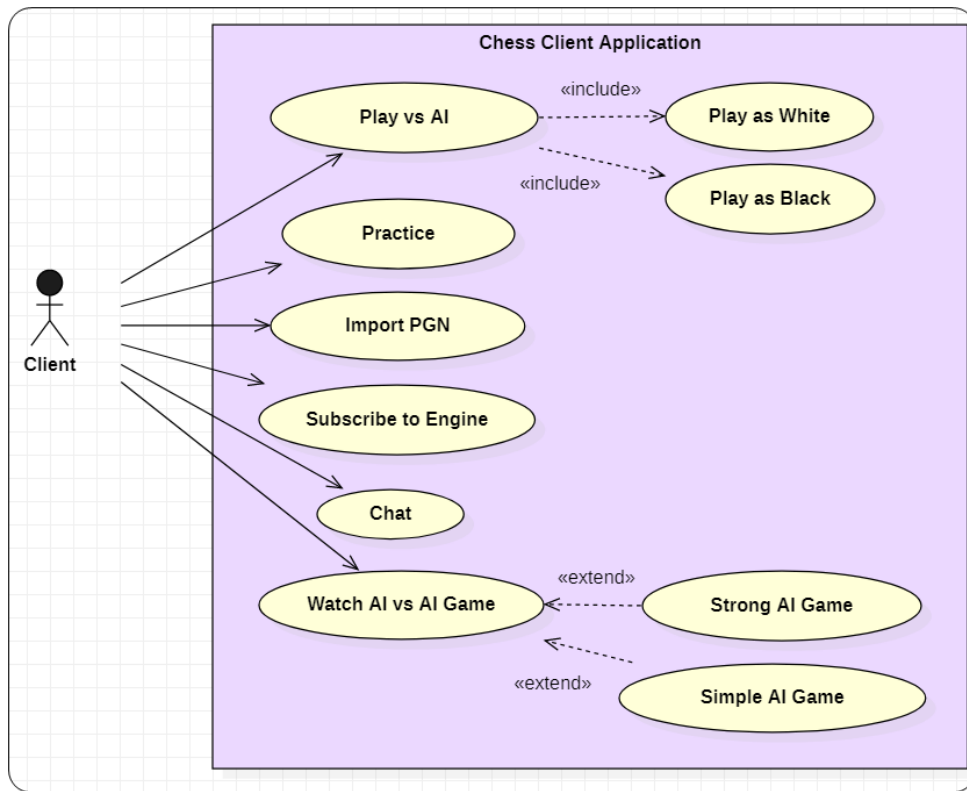


Figure 1.1: Diagrama cazurilor de utilizare pentru aplicația **Client**

Serverul din cadrul proiectului va prelua cereri ale clienților și va implementa un chatroom. În momentul în care un client dorește să se conecteze la server, acesta mai întâi trebuie să introducă un nickname (o poreclă) iar ulterior toți ceilalți clienți vor fi notificați de apariția unui utilizator nou în camera de chat.

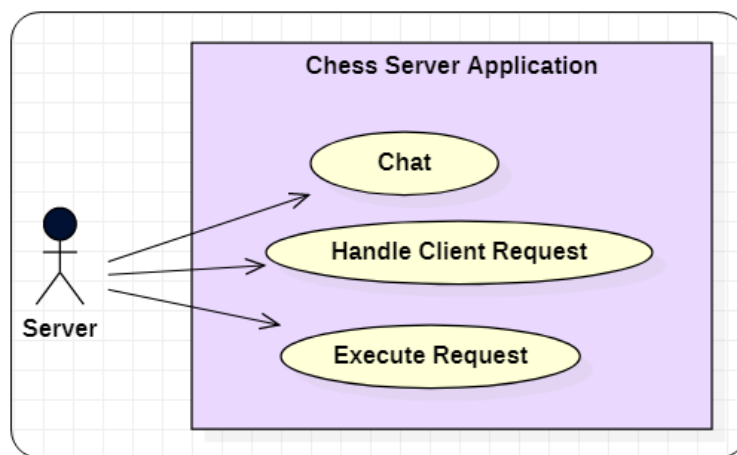


Figure 1.2: Diagrama cazurilor de utilizare pentru aplicația **Server**

```

SERVER = 172.17.153.49
[STARTING] server is starting...
[LISTENING] Server is listening on 172.17.153.49
Nickname of the client is radu
[NEW CONNECTION] ('172.17.153.49', 57182) connected.
[ACTIVE CONNECTIONS] 2
    
```

Figure 1.3: Exemplu de utilizare al serverului

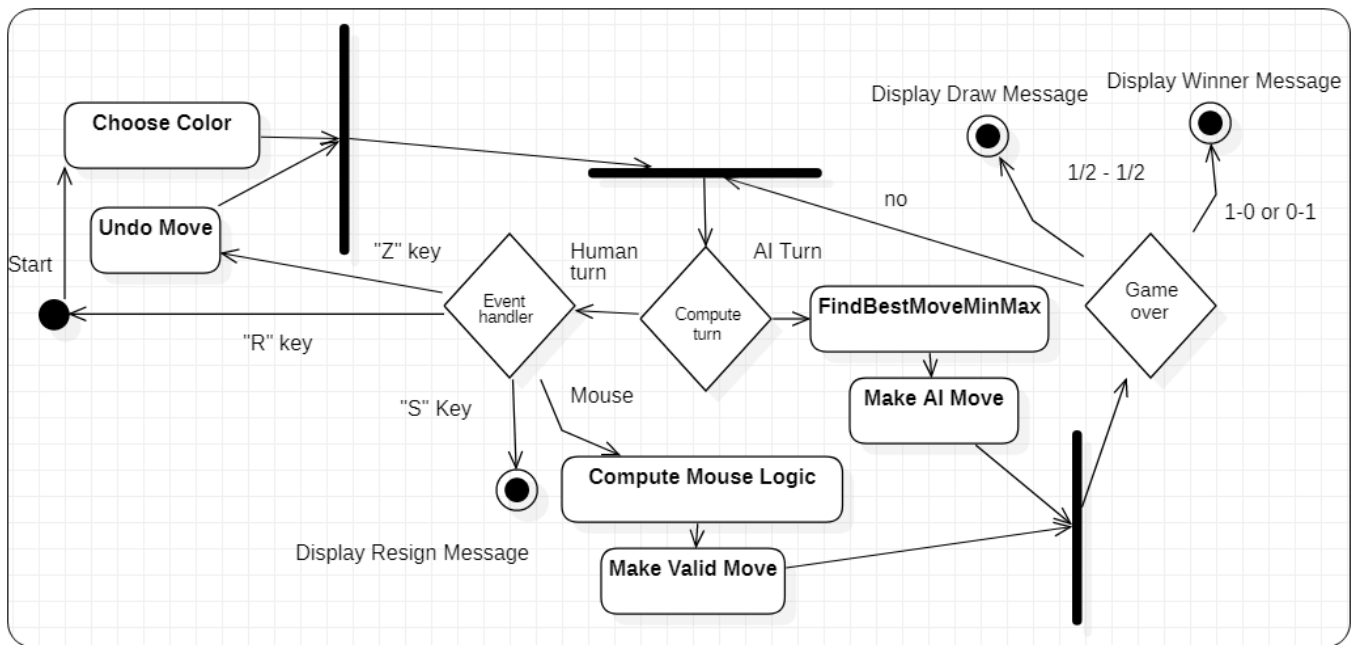


Figure 1.4: Diagrama de activitate a cazului de utilizare : "Human vs AI"

Clientul alege culoarea pieselor cu care dorește să joace, această decizie afectează două variabile **playerOne** și **playerTwo**. Dacă se setează **playerOne** pe **True** și **playerTwo** pe **False** atunci jucătorul uman va avea la dispoziție piesele albe iar calculatorul va juca cu piesele negre. Pentru **playerOne** setat pe **False** și **playerTwo** setat pe **True**, jucătorul uman va avea la dispoziție piesele negre. Când este timpul inteligenței artificiale să realizeze o mutare, se calculează și se returnează cea mai bună decizie după care urmează rândul adversarului. Jucătorul uman poate să refacă ultima mutare, să reînceapă un joc sau să renunțe.

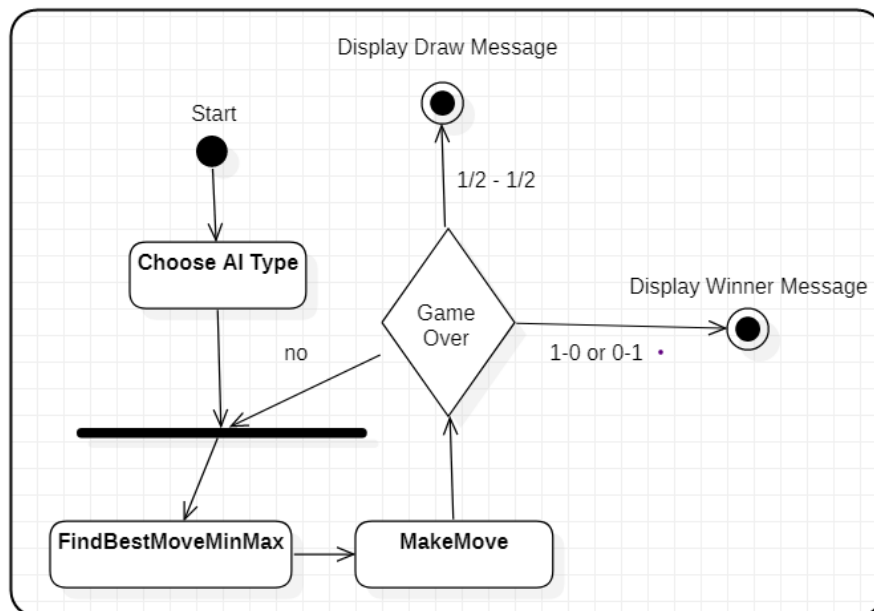


Figure 1.5: Diagrama de activitate a cazului de utilizare : "AI vs AI"

Această diagramă este mai simplă deoarece singura interacțiune a clientului este alegerea tipului de AI. De data aceasta nu se mai așteaptă utilizatorul să realizeze o mișcare, fiecare AI realizând pe rând mutarea calculată de algoritmul MIN-MAX.

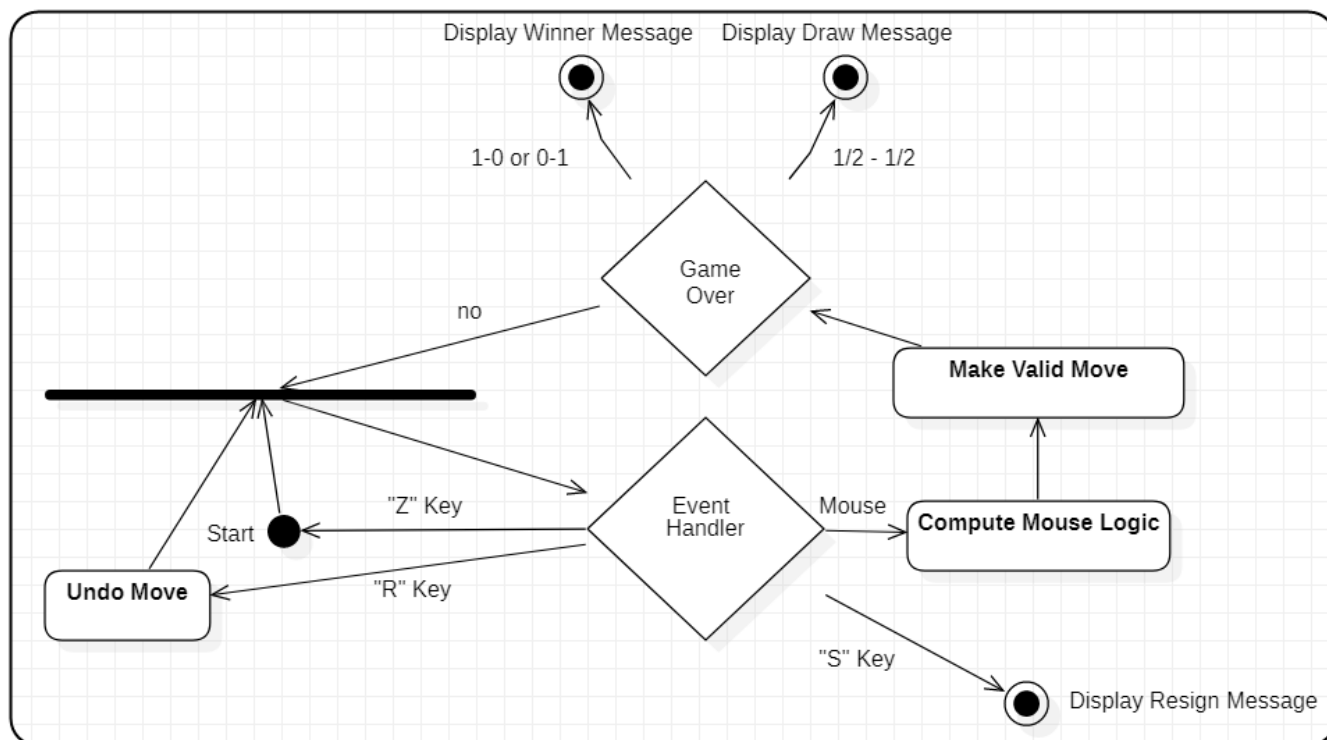


Figure 1.6: Diagrama de activitate a cazului de utilizare : "Practice"

În acest mod de joc nu intervine inteligența artificială iar clientul realizează toate mutările, atât pentru piesele albe, cât și pentru cele negre, cu posibilitatea reîntoarcerii și renunțării în orice moment.

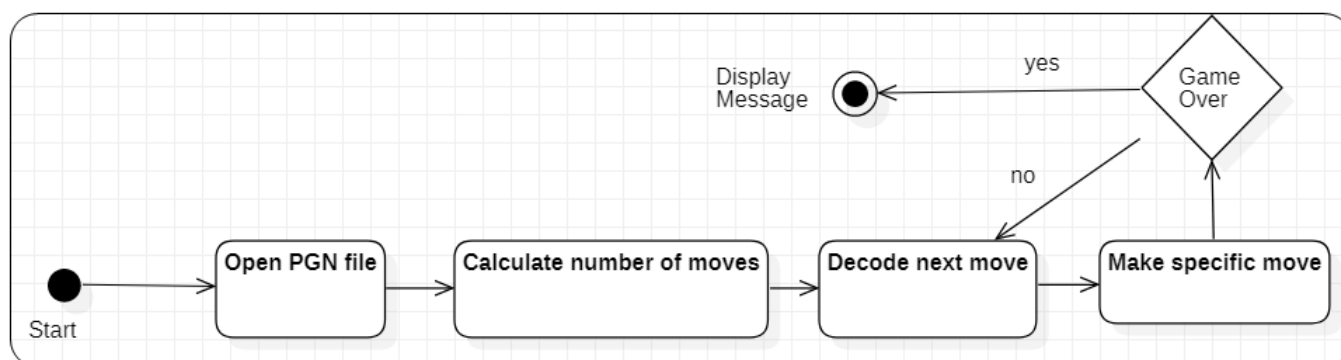


Figure 1.7: Diagrama de activitate a cazului de utilizare : "Import PGN"

Utilizatorul alege fișierul pe care dorește să îl încarce iar programul va decodifica fiecare mișcare (va transforma notațiile algebrice în mutări propriu zise, de exemplu mutarea "e4" reprezintă a 4-a linie și a 5-a coloană de pe masa de șah). Se poate încărca ultimul meci jucat în cadrul aplicației deoarece la finalul oricărui joc, toate mutările se salvează într-un fișier cu format compatibil.

```

1.c4 Nf6 2.Nc3 c5 3.Nf3 e6 4.g3 b6 5.Bg2 Bb7 6.0-0 Be7 7.d4 cxd4 8.Qxd4 d6
9.Bg5 a6 10.Bxf6 Bxf6 11.Qf4 0-0 12.Rfd1 Be7 13.Ne4 Bxe4 14.Qxe4 Ra7 15.Nd4 Qc8
16.b3 Re8 17.a4 Qc5 18.Ra2 Bf6 19.Rad2 Rc7 20.Qb1 Be7 21.b4 Qh5 22.Rc2 Rec8
23.b5 axb5 24.axb5 Rxc4 25.Rxc4 Rxc4 26.Qa2 Qc5 27.Qa8 Rxd4 28.Qxb8+ Bf8
29.Ra1 d5 30.Bf1 Rc4 31.Ra8 Rc1 32.Qe8 d4 33.Ra7 Qf5 34.Ra8 Qc5 35.g4 Qd6
36.Rd8 Qb4 37.Rd7 h6 38.Qxf7+ Kh7 39.g5 Qb1 40.g6+ Qxg6+ 41.Qxg6+ Kxg6 1/2-1/2
  
```

Figure 1.8: Exemplu de fișier care respectă formatul PGN

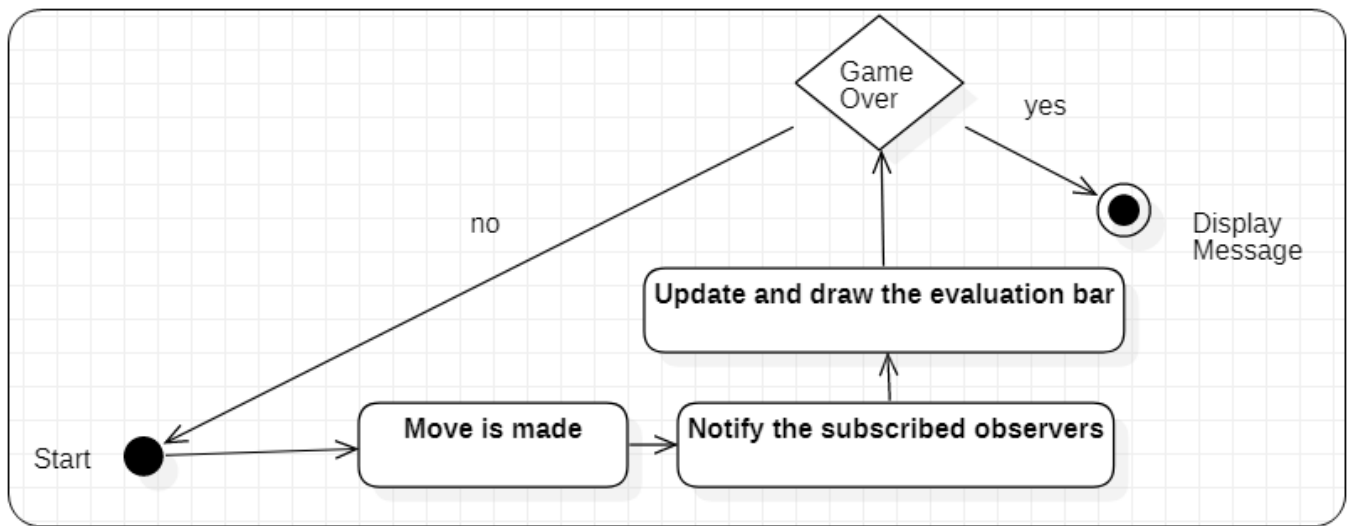


Figure 1.9: Diagrama de activitate a cazului de utilizare : "Subscribe"

Dacă utilizatorul se abonează la motorul de șah al aplicației, apăsând butonul specific aflat în dreapta meniului principal, acesta va fi notificat pe interfață că poate beneficia de o **bară de evaluare** specifică oricărui motor de șah. Aceasta bară ilustrează scorul poziției curente de pe tabla de șah (dacă bara indică un scor pozitiv atunci jucătorul pieselor albe are o poziție mai bună decât jucătorul pieselor negre și șanse mai mari de victorie implicit, în caz contrar negrul deține avantajul). Adicional se va afișa în dreapta ferestrei aplicației istoricul jocului, indiferent dacă utilizatorul este abonat sau nu. Pentru notificarea abonaților cu scorul curent și bara de evaluare s-a folosit șablonul de proiectare comportamental **Observer**.

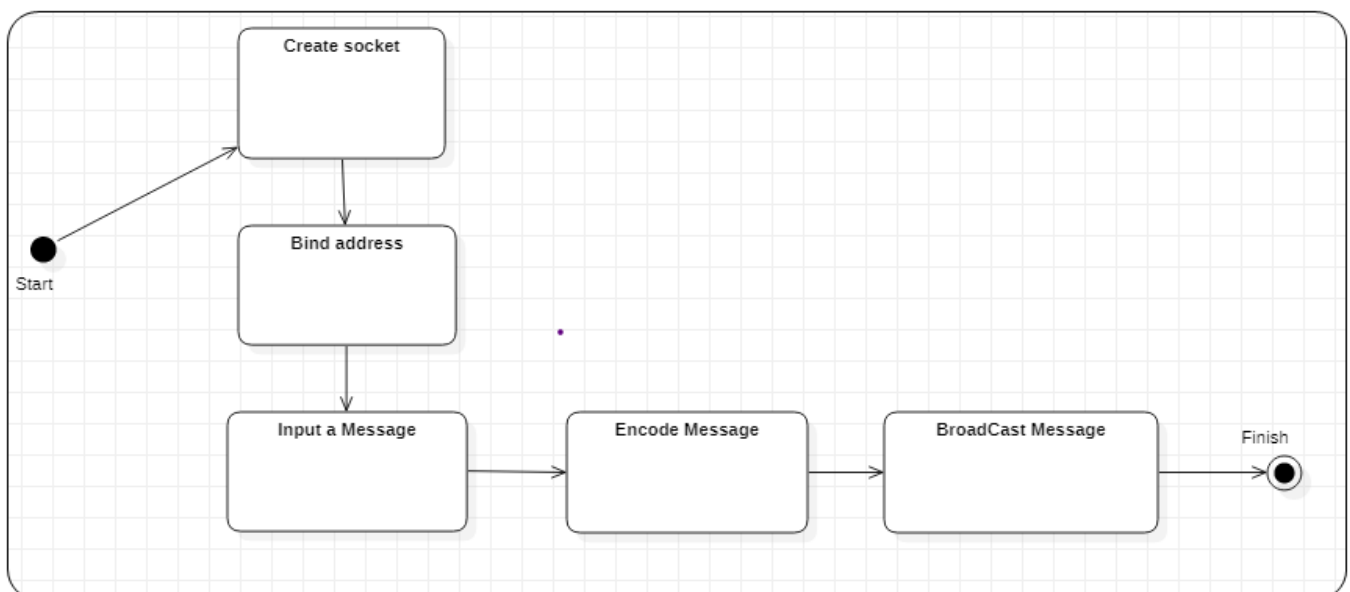


Figure 1.10: Diagrama de activitate a cazului de utilizare (Server) : "Chat"

Proiectare

În această fază se realizează diagrama de clase corespunzătoare proiectului, după care se vor realiza diagramele de secvență corespunzătoare tuturor cazurilor de utilizare.

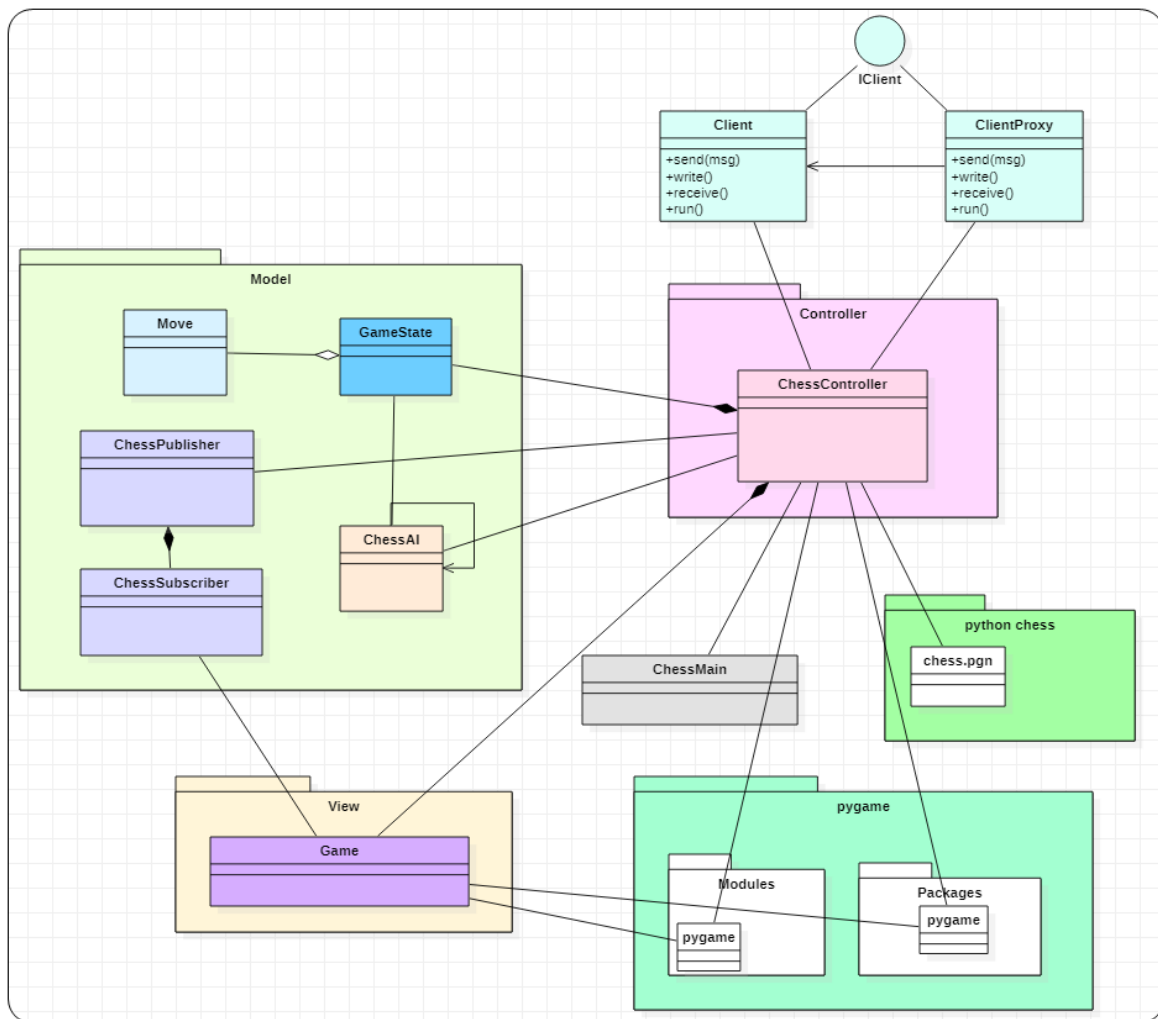


Figure 2.1: Diagrama de clase (simplificată)

Mesajele din diagramele de secvență reprezintă metode iar blocul de "loop" indică faptul că aplicația este în viață până când utilizatorul dorește să o închidă și oferă posibilitatea reîntoarcerii în meniu pentru schimbarea cazului de utilizare, prin metoda "runGame". Pentru cazul de utilizare "AI vs AI", la fiecare mutare, controller-ul trimite un mesaj clasei ChessAI, solicitând calculul celei mai puternice mutări posibile. Mesajul este transmis mai departe la modelul propriu zis al aplicației, care realizează mutarea, schimbând starea jocului (matricea pieselor). De asemenea modelul răspunde la mesaj, actualizând lista tuturor mișcărilor valide pentru starea curentă. Odată realizată cu succes mutarea, se desenează pe ecran masa de șah.

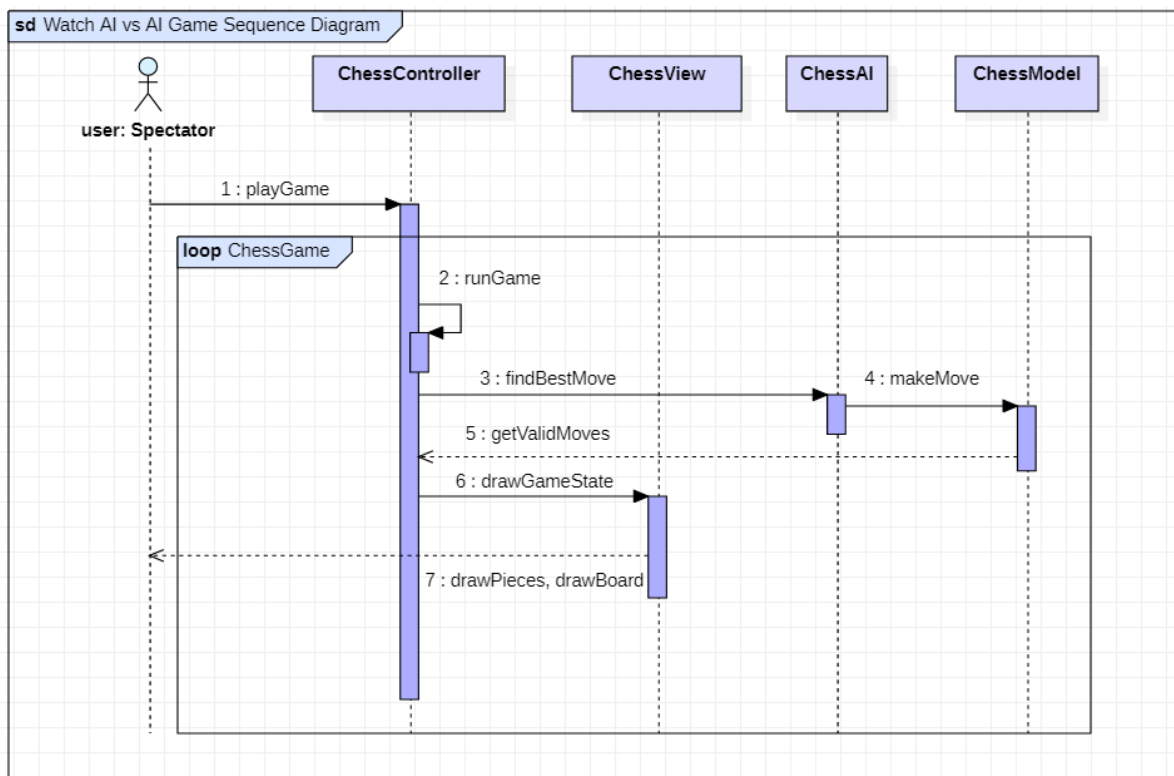


Figure 2.2: Diagrama de secvență a cazului de utilizare : "AI vs AI"

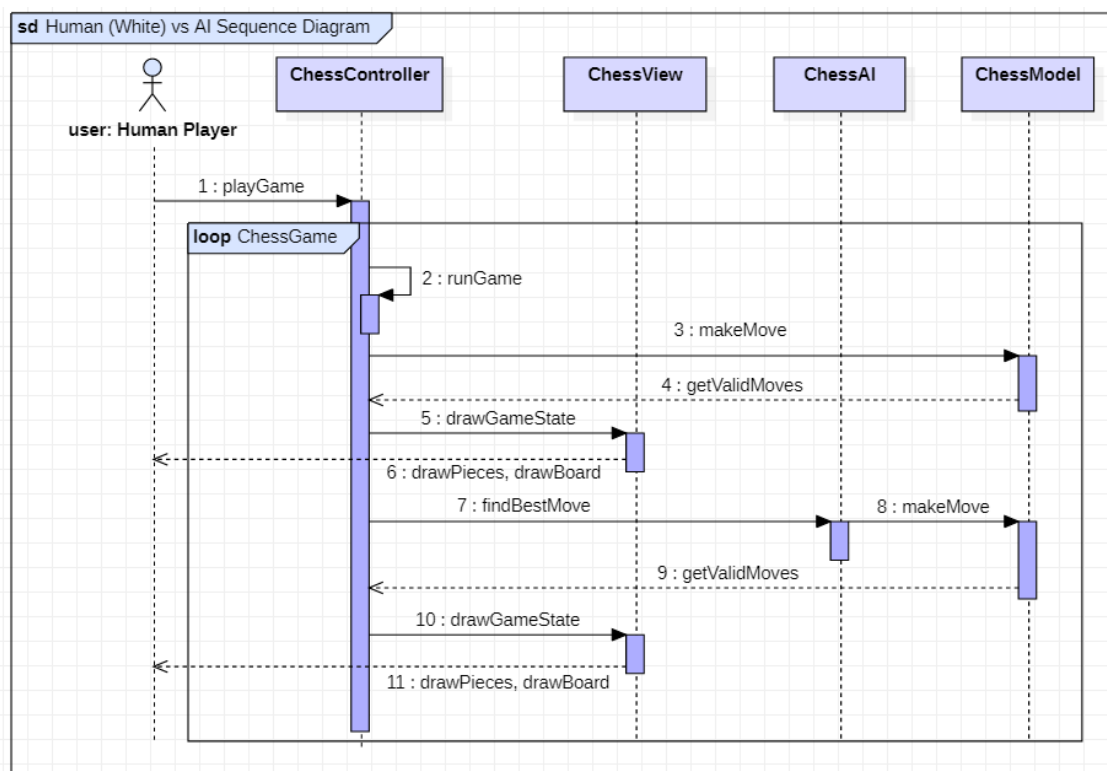


Figure 2.3: Diagrama de secvență a subcazului de utilizare : "White vs AI"

Pentru acest subcaz de utilizare, AI-ul realizează prima mutare deoarece are piesele albe. Când este rândul clientului, mesajul se transmite direct către clasa model și nu către ChessAI.

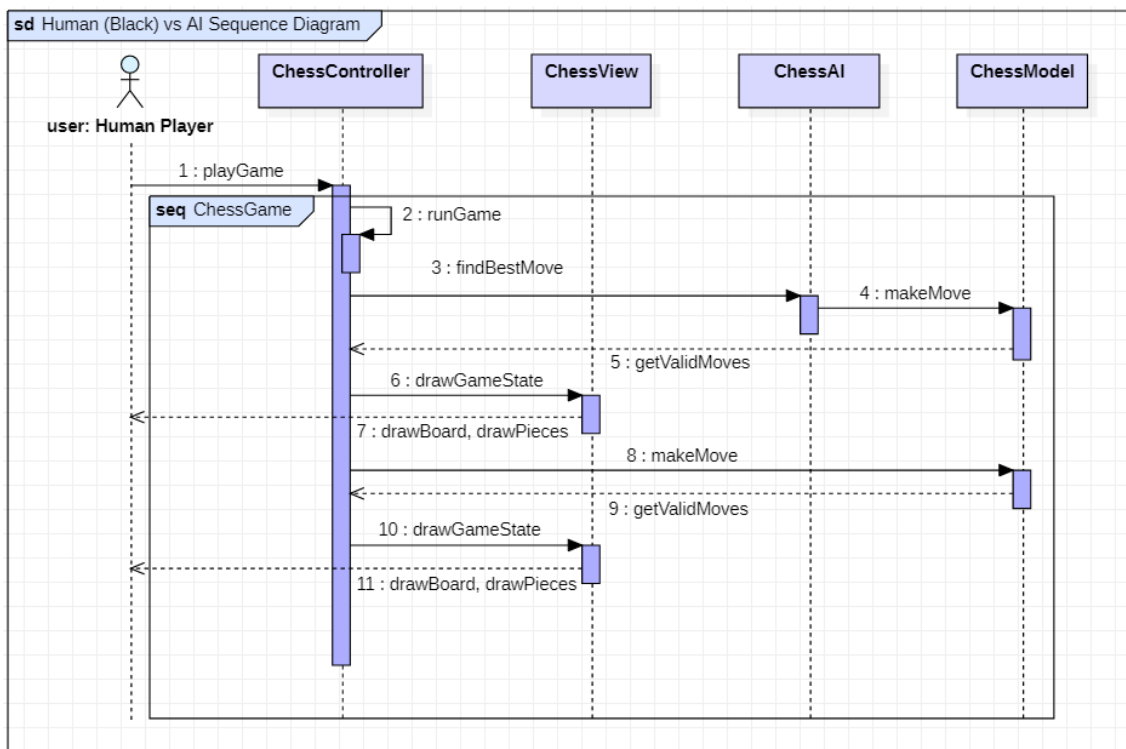


Figure 2.4: Diagrama de secvență a subcazului de utilizare : "AI vs Black"

Analog subcazului precedent, numai că de data aceasta clientul deține piesele albe.

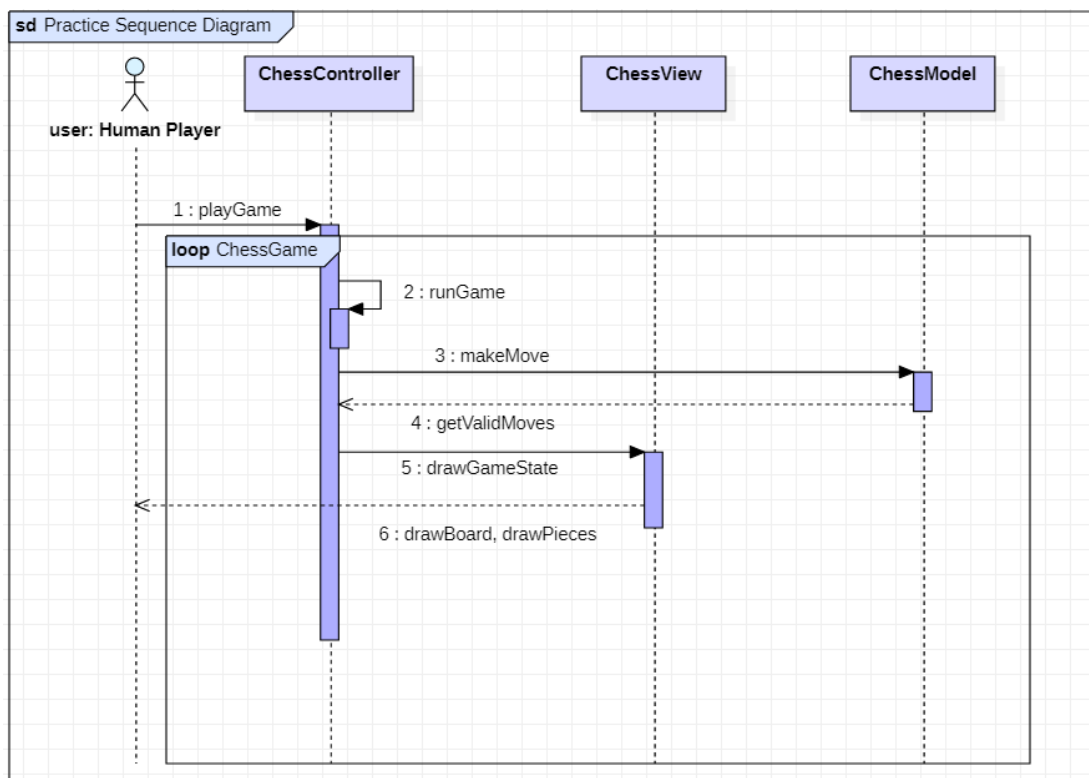


Figure 2.5: Diagrama de secvență a cazului de utilizare : "Practice"

Analog cazurilor de utilizare precedente, numai că de data aceasta nu se transmite mesaje clasei ChessAI.

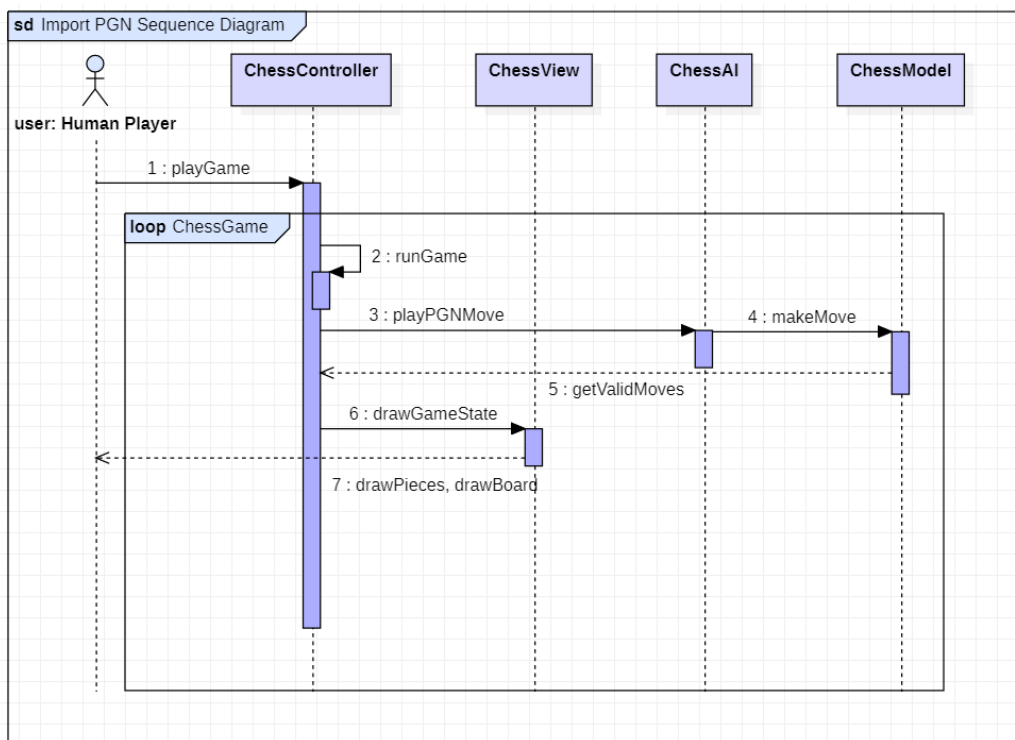


Figure 2.6: Diagrama de secvență a cazului de utilizare : "Import PGN"

Pentru acest caz de utilizare nu se mai folosește algoritmul MIN-MAX, ci se decodifică și se realizează implicit mutarea din fișier, cu ajutorul metodei "playPGNMove"

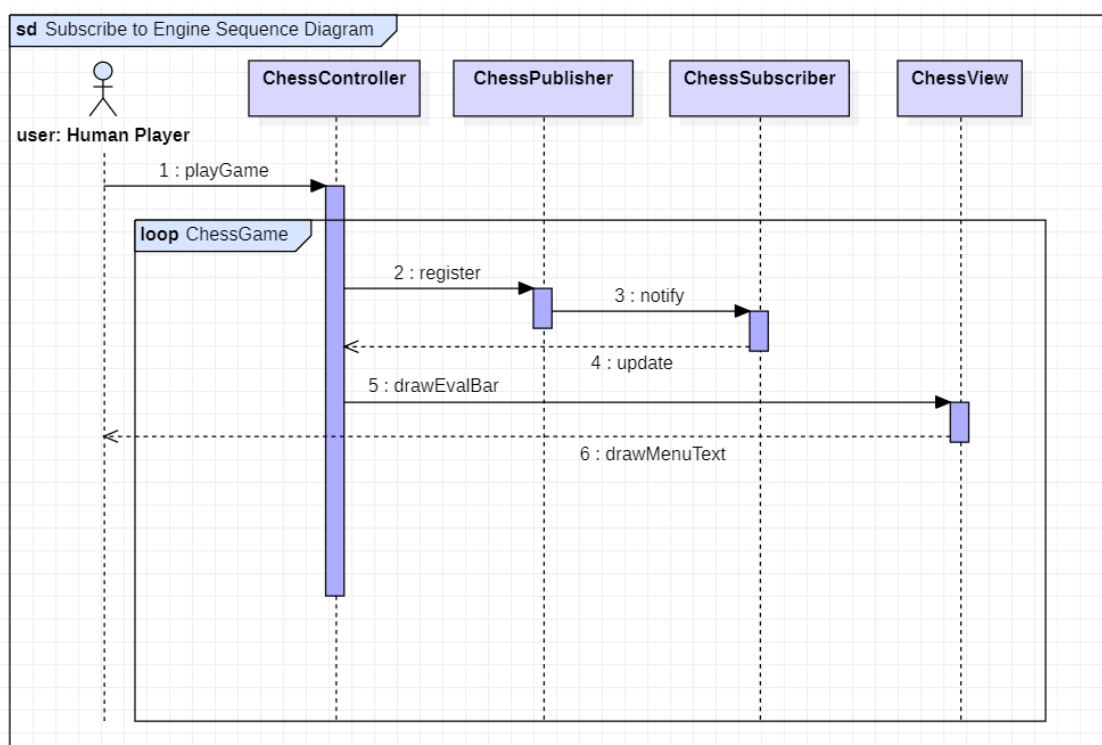


Figure 2.7: Diagrama de secvență a cazului de utilizare : "Subscribe"

Metoda register adaugă în lista abonatilor din clasa ChessPublisher un nou utilizator iar în momentul în care are loc o nouă mutare, se notifică toți abonații și se actualizează bara de evaluare după scorul curent indicat de AI.

Implementare

În această fază se scrie codul pentru îndeplinirea tuturor funcționalităților menționate. Limbajul de programare folosit în realizarea acestui proiect este **Python**. Motivul principal pentru aceasta alegere este faptul că s-a preferat simplitatea peste eficiență. Python este un limbaj interpretat, în timp ce C de exemplu este un limbaj compilat. Codul interpretat este întotdeauna mai lent decât codul direct al mașinii, deoarece este nevoie de mai multe instrucțiuni pentru a implementa o instrucțiune interpretată decât pentru a implementa o instrucțiune reală a mașinii.

Se așteaptă, în general, ca programele Python să ruleze mai lent decât programele **Java** sau **C**, dar durează mult mai puțin timp pentru a se dezvolta. Programele Python sunt de obicei de **3-5** ori mai scurte decât programele Java echivalente.

Pentru desenarea tablei de șah sau a barei de evaluare în fiecare moment și afișarea mesajelor de victorie sau egal după încheierea partidei, dar și pentru interpretarea evenimentelor care apar de la momentul pornirii aplicației, s-a folosit setul de module și clase oferite de **Pygame**. Acesta adaugă funcționalități noi pe lângă cele oferite de biblioteca SDL. Principalul avantaj în folosirea acestui set este faptul că nucleul este păstrat simplu și lucruri suplimentare precum bibliotecile GUI și efectele sunt dezvoltate separat în afara lui **Pygame**.



Pentru parsarea și parcurgerea mai eficientă a fișierelor care respectă formatul PGN, s-a utilizat librăria Python chess.pgn, care oferă numeroase funcționalități (vezi <https://python-chess.readthedocs.io/en/latest/pgn.html>).

Interfața grafică inițială

Folosind modulele implementate de **Pygame**, s-a realizat o interfață care oferă utilizatorului posibilitatea de a alege modul de joc pe care acesta dorește să-l joace, cu posibilitatea revenirii înapoi la meniul principal (apăsarea tastei "**R**") (Instrucțiunile sunt afișate în meniul principal, în partea dreaptă). Modurile disponibile sunt (în ordinea din imaginea de mai jos) :



Figure 3.1: Interfata grafică inițială.

1. Vizualizarea unui joc între doi agenți mini-max (adâncime 2)
2. Începerea unui joc împotriva unui agent mini-max (adâncime 2) având piesele albe.
3. Începerea unui joc împotriva unui agent mini-max (adâncime 2) având piesele negre.
4. Începerea unui joc în care utilizatorul controlează toate piesele (exersare).
5. Vizualizarea unui joc între doi agenți mini-max (adâncime 3)
6. Începerea unui joc împotriva unui agent mini-max (adâncime 3) având piesele albe.

În cazul în care utilizatorul alege un mod în care trebuie să mute piesele și nu este spectator (2, 3, 4, 6), acesta are posibilitatea de a anula ultima mutare (undo) cu ajutorul tastei "Z" și posibilitatea de a renunța (resign), cu ajutorul tastei "S". În continuare se prezintă câteva rezultate inițiale:



(a) Renunțare



(b) Mat oferit de AI



(c) Meci egal



(d) Victorie alb

Figure 3.2: Exemple meciuri AI vs AI

Interfața grafică finală

Interfața inițială s-a rafinat, utilizatorul având acum posibilitatea de a selecta limba dorită pentru utilizarea aplicației (dintr-un total de 4 limbi). Stringurile care reprezintă mesajele afișate pe interfață sunt stocate în fișierul `display.xml`. De asemenea, utilizatorul mai are posibilitatea de a se abona la motorul de șah al aplicației, apăsând butonul specific aflat în dreapta meniului principal. Odată abonat, acesta va fi notificat pe interfață că poate beneficia de o **bară de evaluare** specifică oricărui motor de șah. Aceasta bara ilustrează scorul poziției curente de pe tabla de șah (dacă bara indică un scor pozitiv atunci jucătorul pieselor albe are o poziție mai bună decât jucătorul pieselor negre și șanse mai mari de victorie implicit, în caz contrar negrul deține avantajul). Adicional se va afișa în dreapta ferestrei aplicației istoricul jocului, indiferent dacă utilizatorul este abonat sau nu. Pentru notificarea abonaților cu scorul curent și bara de evaluare s-a folosit șablonul de proiectare comportamental **Observer**.



(a) Meniul principal în germană



(b) Meniul principal în rusă

Scurt tutorial

Se intră în aplicația Server, se deschide un terminal (dacă se lucrează în Pycharm), altfel se deschide un terminal în command prompt și se alege calea către directorul proiectului Server și se deschide serverul folosind comanda `python Server.py`, după care se deschide aplicația client în același mod folosind comanda `python Client.py`.



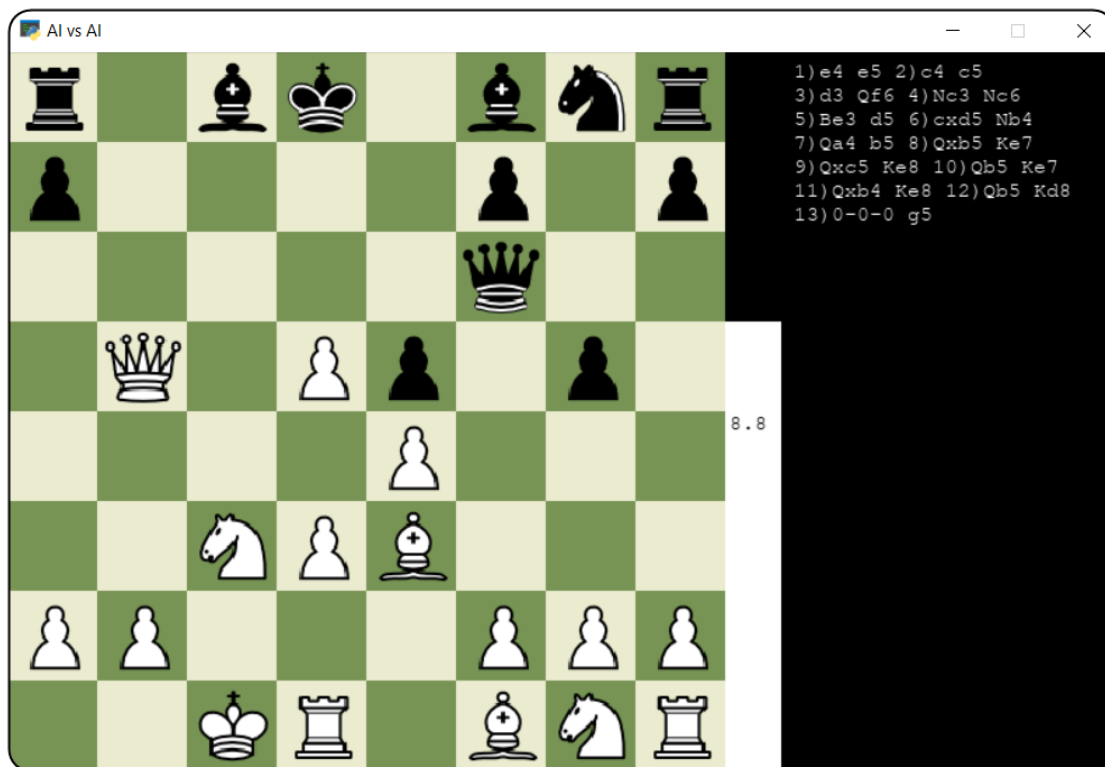


Figure 3.3: Pozitie favorabilă pieselor albe

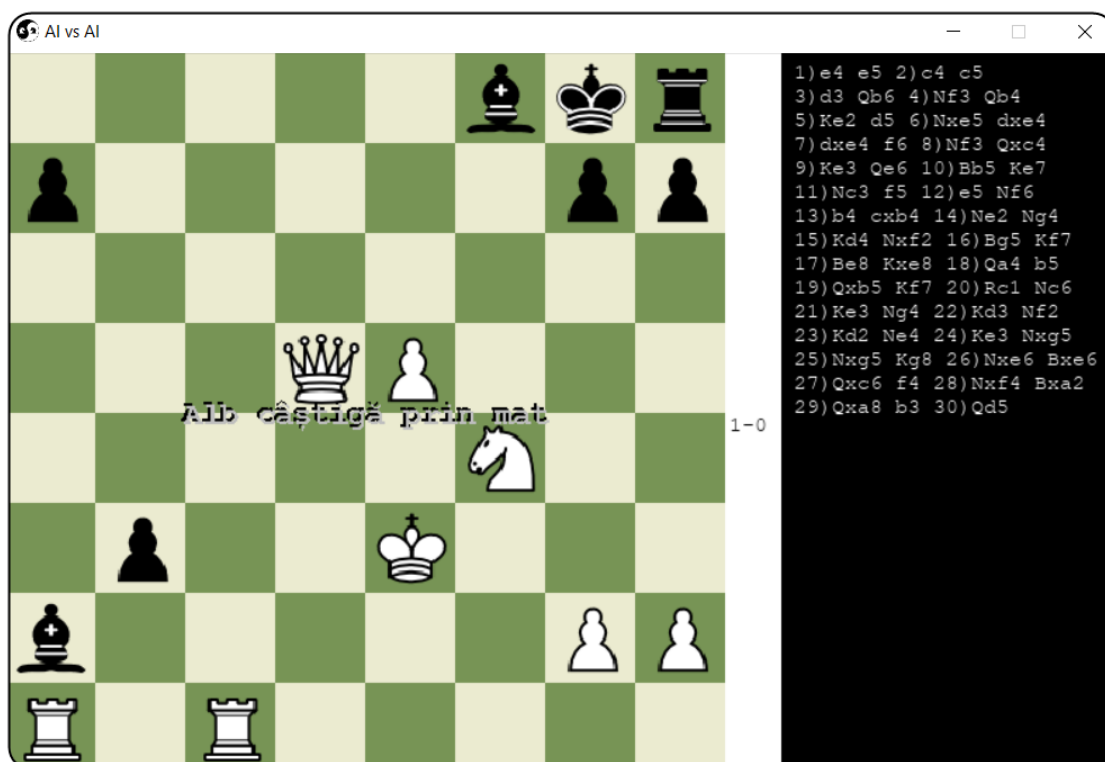


Figure 3.4: Alb câștigă prin mat

* * *

Intelligent Systems Group

