

Project Chess

21.01.2021

Îndrumător:

dr. ing. Daniel Morariu

Student:

Caraganciu, Alexandrin

(224/2)

Istoric Versiuni

Data	Versiune	Descriere	Autor
03.12.2020	0.1	<ul style="list-style-type: none"> • Meniul principal: Butonul de start de joc, butonul de căutare de joc, butonul de ieșire. • Design de start. • Noua fereastra pentru joc, după apăsare butonului de start • Ieșirea din program după apăsarea butonului de ieșire 	Alex
11.12.2020	0.2	<ul style="list-style-type: none"> • Grid pentru fereastra de joc (GridPane cu 2 coloane: 1 coloana pentru tabla, 2 coloana pentru butoane) • Selectarea culorilor pentru pătrate din tablou • Desenarea tablei pe ecran cu funcția de resize automată pentru pătrate => tabla este resizable pentru orice ecran 	Alex
12.12.2020	0.3	<ul style="list-style-type: none"> • Formatarea Gridului a ferestrei (HBox, care tine în el un GridPane pentru tabla și VBox pentru butoane) • Îmbunătățirea sistemului de resize. Acum tabla își păstrează raportul laturilor a celulelor. • Setarea imaginii a figurii în tablou 	Alex
12.12.2020	0.4	<ul style="list-style-type: none"> • Bug fixat: După maximizarea ferestrei cu butonul respectiv tabla nu schimba mărimea sa • Schimbat sistem de plasarea imaginii, acum se utilizează imagini vectoriale svg, în loc de imagini png. Așa se poate de salvat timpul pornirii programului, pentru ca noi nu mai avem nevoie să încărcăm imaginea și păstrăm loc, pentru ca imaginile svg pot fi modificate utilizând CSS (culoare, mărimi etc.). • Redesign complet a aplicației • Scoaterea generarea tablei în clasa aparte, care se numește Chessboard. 	Alex
12.12.2020	0.5	<ul style="list-style-type: none"> • Clas separat pentru fiecare piesa de șah • Tipul de date enum pentru culorile pieselor • Matricea "pieces" pentru setarea pieselor pe tabla • Citirea din matricea și setarea pieselor pe tabla 	Alex
12.12.2020	0.5.1	<ul style="list-style-type: none"> • Clasa PiecesPosition pentru setarea inițială a pieselor pe tabla și prelucrarea pieselor mișcate și rămase pe ecran 	Alex

13.12.2020	0.7	<ul style="list-style-type: none"> ● Implementarea mișcărilor pieselor cu drag and drop ● Setarea marginii colorate deasupra celulei selectate ● Nu se poate de pus piesa deasupra piesei sale (în viitor cu ajutorul verificării acestea vor fi implementate mișcărilor figurilor după regulile șahului) ● Tranziție din metode statice în moștenire unde posibil ● Adăugarea sunetului la mișcarea pieselor ● Scoaterea metodelor pentru resize a tabloului în clasa Chessboard 	Alex
04.01.2021	-	<ul style="list-style-type: none"> ● Model de server/client ● Transmiterea mesajelor între clienți 	Alex
05.01.2021	0.9	<ul style="list-style-type: none"> ● Implementarea clasei Player (clientul șahului) ● Serverul începe lucru doar atunci, când ambii jucători s-au conectat ● Serverul nu conectează mai mult de 2 jucători ● Forma de selectarea culorii pieselor de către jucător ● Jucător poate face doar o mișcare după ce transmite mișcarea altui jucător ● Jucător nu poate face mișcarea cu figura de culoarea altuia jucător ● Extragerea tipului de date PieceColor într-un fișier aparte pentru acces către el din mai multe clase simultan ● Butonul "Find Game" acum conectează jucător către joc deja creat. 	Alex

07.01.2021	1.0.0	<ul style="list-style-type: none"> ● Adăugarea unui label “gameInfo” pentru afișarea informațiilor utile jucătorilor ● Dacă al doilea jucător încă nu s-a conectat, atunci primul jucător nu poate face mișcări, chiar dacă are piese albe ● În gameInfo se afișează dacă este mișcarea ta ● Acum mișcarea ta se termina doar după apăsarea tastei “End Turn”. ● Crearea clasei noi “Move” și crearea proprietății move de clasa Move în clasa Chessboard. Clasa aceasta păstrează coordonatele și start și sfârșit a mișcării a piesei ● Reformatarea metodei “sendMove” în “sendMessage” a clasei Player pentru a optimiza numărul de metode ● Tabla de șah se rotește în funcție de culoarea a pieselor selectată ● Dacă jucător închide fereastra de joc, atunci el este automat deconectat de la server, ca și jucător, care a rămas în joc. Jucător rămas obține mesaj corespunzător ● Implementarea metodelor pentru butoane “Check”, “Checkmate” și “Surrender” ● Acum jucător poate apăsa “End Turn”, “Check” și “Checkmate” doar dacă el a făcut, dar încă nu a terminat mișcarea sa ● Butonul “Surrender” poate fi apăsat oricând de orice jucător. Apăsarea butonului automat setează jucător care a apăsat buton ca jucător învins și alt jucător ca învingător. Jucătorii pierd posibilitatea de a face mișcări și sunt deconectați de la server. ● Când jucători sunt deconectați de pe server butoanele “End Turn”, “Check”, “Surrender” și “Checkmate” sunt blocate și un buton nou “Close Game” apare, care închide fereastra de joc și întoarce jucător pe fereastra cu meniul principal, ce îl dă posibilitate de a începe un joc nou! ● Schimbarea metodei de mișcarea a pieselor. Acum celulă cu piese selectate obține culoare roșie, pentru a arată ca din ea se va mișca piesa. Când mouse se afla deasupra altor celule ei obțin culoare alba, pentru a arata la ce celula se uita cursorul. 	Alex
08.01.2021	1.0.5	<ul style="list-style-type: none"> ● Bug corectat. Acum dacă celulă unde se muta o piesa este celula unde piesa si era border roșu dispăre și nu rămână pentru o infinitate ● Posibilitatea de a selecta culoare pieselor random ● Crearea funcțiilor pentru mișcarea pieselor în partea și vizuala și în partea de cod, optimizând și micșorând cod 	Alex

		<ul style="list-style-type: none">● Realizarea de castling cat lung, atât și scurt	
15.01.2021	1.1.0	<ul style="list-style-type: none">● Când pawn ajunge la sfârșitul tablei, apare o fereastră modală, care cere jucător să-și aleagă o piesa, care va înlocui pe pawn	Alex
16.01.2021	1.1.1	<ul style="list-style-type: none">● Bug fixat: Acum fereastră modală pentru selectarea piesei în locul pionului nu poate fi închisă fără a selecta o piesa.● Bug fixat: Acum dacă event drag & drop se termină în afara tablei de șah border de selecție de pe celulă de start va dispări● Bug fixat: Acum după rocada sau schimbarea piesei și apăsarea butonului “Check” sau “Checkmate” aplicația nu mai dă eroare și prelucrează situația dată corect	Alex
18.01.2021	1.1.2	<ul style="list-style-type: none">● Bug fixat: Acum dacă este conectat doar un singur jucător și el apasă pe butonul “Surrender” sau se deconectează, server în deconectează și lucrează cum este exceptat de el(așteaptă doi jucători noi)	Alex

Cuprins

Specificarea cerințelor software	8
Introducere	8
Obiective	8
Definiții, Acronime și Abrevieri	8
Tehnologiile utilizate	9
Cerințe specifice	9
Funcționalitate. Mișcarea pieselor pe tabla de șah și terminarea mișcării	9
Descriere	10
Fluxul de evenimente	10
Fluxul de bază	10
Fluxuri alternative	11
< Primul flux alternativ >	11
Pre-condiții	11
Post-condiții	11
Funcționalitate. Obținerea mesajelor de pe server de către jucător	12
Descriere	12
Fluxul de evenimente	12
Fluxul de bază	12
Fluxuri alternative	12
< Primul flux alternativ >	12
Pre-condiții	13
Post-condiții	13
Funcționalitate. Server	14
Descriere	14
Fluxul de evenimente	14
Fluxul de bază	14
Fluxuri alternative	14
< Primul flux alternativ >	14
Pre-condiții	14
Post-condiții	14
Funcționalitate. Start de joc	15
Descriere	15
Pre-condiții	15
Post-condiții	15
Implementare	16
Diagrama de clase (Joc)	16
Diagrama de clase (Server)	19
Descriere detaliată	20

Meniul principal:	20
Fereastră de joc:	21
Server:	24
Bibliografie	26

1 Specificarea cerințelor software

1.1 Introducere

Aplicația reprezintă un joc de șah în rețea. Scopul aplicației era în a crea un instrument pentru ca două persoane să se joace șah, aflându-se oriunde. Aplicația nu a avut ca scop principal verificarea ca jocul să se joace după regulile șahului, dar să creeze condiții pentru jucători să-și arate ce mișcări ei au făcut.

1.1.1 Obiective

- Afișarea pieselor pe tabla, poziția cărora poate fi ușor regulată de către programator
- Mișcarea pieselor pe tabla de șah
- Rocada
- Selectarea piesei noi, dacă un pion a ajuns la partea opusă a tablei
- Transmiterea mișcărilor efectuate prin rețea
- Salvarea și încărcarea jocului, dacă jucătorii nu mai au timp pentru a juca
- Abilitatea serverului de a conduce mai multe jocuri simultan
- User friendly design a aplicației
- Urmarea a cui din jucători este mișcarea la moment dat
- Supravegherea ca jucător, care joacă cu piese negre să nu are posibilitatea să joace cu piese oponentului
- Codificarea realizată pentru mesaje transmise
- Rotirea tablei de joc, ca piesele de culoarea jucătorului să înceapă din partea de jos a tablei
- Conectarea jucătorilor într-un joc
- Prelucrarea situației dacă unul din jucători, sau amândoi o sa iasă din aplicația în timp ce joc nu era terminat
- Unică situația de a termina joc “legal” este de a apăsa butonul “surrender”
- Undo ultimul pas
- Posibilitatea de a juca pe un singur calculator

1.1.2 Definiții, Acronime și Abrevieri

Metoda de codificarea mesajelor. Mesaj între client server constă din mai multe părți importante. Există trei tipuri de mesaje: mesaje de conectare, mesaje de mișcare și mesaje de deconectare.

Mesaje de conectare sunt: <Culoarea selectată> sau “connected”

- Culoarea selectată se transmite către server, ce îl ajute să stabilească culoarea corectă jucătorului, care se va conecta. (detalii în partea de funcționalități: obținerea mesajului, server)
- Culoarea selectată se transmite către jucător pentru ca el să știe ce culoare el va avea.
- “connected” se transmite jucătorului, care a început joc în moment, când vreun jucător s-a conectat la el.

Mesaje de mișcare: au în esența coordonatele de start a mișcării și unde piesa s-a mutat.

!IMPORTANT! mesajul NU conține precizat piese, care s-a mișcat, nici culoarea, care a făcut mișcarea, deoarece este logic, că partea care obține mesaj nu a făcut mișcarea aceasta.**!IMPORTANT!**
[“Check” | “Checkmate”][“shortCastling” | “longCastling” | <denumirea piesei selectate>] x y x1 y1.

- Check sau Checkmate apare în moment ce jucător a apăsă tasta de Check sau Checkmate respectiv.
- shortCastling sau longCastling spune jucătorului că a fost efectuată rocada scurtă sau lungă respectiv și nu o mișcarea simplă.

- denumirea piesei selectate apare când jucător a selectat o nouă piesa din fereastră modală în moment când pionul lui a ajuns la sfârșitul tablei. Ea ajută jucătorului, care a obținut mesaj să înțeleagă ce piesa a fost selectată
- $x\ y\ x1\ y1$ - coordonatele de unde a fost mutată piesa și unde ea a fost mutată

Mesaje de deconectare: “surrender” | “disconnected”

- “surrender” se transmite la moment când jucătorul a apăsă butonul respectiv
- “disconnected” se transmite la moment când jucătorul a închis fereastra în timp ce joc nu era terminat.

P.S. mai multe detalii în partea cu funcționalități

În aplicația există câteva codificări a variabilelor, dacă variabilă este de tip boolean, ea are denumirea care se începe de “is...” sau “was...”. Toate elementele de tip Button se termină cu cuvântul “...Button”. Toate metode, care reprezintă eveniment au la sfârșitul denumirii “...Handler”

În scheme bloc:

- *cerc* înseamnă tranziție undeva, sau marcarea de început a schemei.
- *romb* înseamnă un if sau dacă sunt posibile mai multe variante unde poate apăsa utilizator
- *trapezie* înseamnă un input ori acțiunea utilizatorului
- *dreptunghi* orice acțiune făcută de către program

1.1.3 Tehnologiile utilizate

- *IntelliJ Idea* - IDE pentru a dezvolta aplicații în Java.
- *Javafx* - platformă construită pe baza de Java pentru a crea aplicații pentru sisteme de operare diferite.
- *Color Mania* - aplicația pentru a selecta culori de pe ecranul calculatorului și a obține codul lor.
- *Scene Builder* - aplicația pentru a ușura redactarea părții vizuale a aplicației scrise utilizând Javafx.
- *Google Docs* - pentru redactarea și scrierea a documentației.

1.2 Cerințe specifice

- Afișarea pieselor pe tabla, poziția cărora poate fi ușor regulată de către programator
- Mișcarea pieselor pe tabla de șah
- Roca
- Selectarea piesei noi, dacă un pion a ajuns la partea opusă a tablei
- Transmiterea mișcărilor efectuate prin rețea
- User friendly design a aplicației
- Urmarea a cui din jucători este mișcarea la moment dat
- Supravegherea ca jucător, care joacă cu piese negre să nu are posibilitatea să joace cu piese oponentului
- Codificarea realizată pentru mesaje transmise - pentru ca server și client (jucător) instant să înțeleagă ce reprezintă mesaj primit
- Rotirea tablei de joc, ca piesele de culoarea jucătorului să înceapă din partea de jos a tablei - dacă jucător joacă cu piese negre, atunci tabla va fi rotită așa ca poziția inițială a pieselor să fie în partea de jos a tablei (mai aproape de jucător)
- Conectarea jucătorilor într-un joc
- Prelucrarea situației dacă unul din jucători, sau amândoi o să iasă din aplicația în timp ce joc nu era terminat
- Unică situația de a termina joc “legal” este de a apăsa butonul “surrender”

2 Funcționalitate. Mișcarea pieselor pe tabla de șah și terminarea mișcării

2.1 Descriere

Mișcarea pieselor pe tabla de șah este realizată cu ajutorul metodei drag & drop. Terminarea mișcării se face

2.2 Fluxul de evenimente

2.2.1 Fluxul de bază

- 1) Pentru a porni metoda drag & drop și a începe mișcarea piesei utilizator trebuie să apese pe piesa dorită și să tragă mouse-ul, fără să da drumul lui click, pe poziția dorită. În acest moment programul obține coordonatele celulei selectate și verifică dacă piesa selectată este de culoare jucătorului, dacă la momentul dat este mișcarea jucătorului și dacă pe poziția selectată există o piesa. Dacă toate condițiile erau îndeplinite, atunci programul salvează în clipboard coordonatele de start și adaugă celulei selectate clasa "movingPiece", care adaugă celulei un border roșu ca jucător să înțeleagă de unde el a început mișcarea piesei.
- 2) Când mouse în proces de drag & drop intră în alta celulă, el adaugă celulei curente o clasa "bordered-move", dacă celulă curentă nu are clasa "movingPiece", care adaugă celulei curente un border alb, ca utilizator să înțeleagă asupra cărei celule el se afla momentan.
- 3) La plecare dintr-o celulă, dacă ea are clasa "bordered-move", ultima se șterge
- 4) La moment când utilizatorul dă drumul lui click programul scoate datele păstrate în clipboard, anume coordonatele de start (de unde s-a început mișcarea) în variabile x , y și scoate coordonatele celulei curente în variabile $x1$, $y1$. Șterge celulei de start clasa "movingPiece". Dacă în celulă de sfârșit este deja o piesa și ea are culoare asemănătoare cu culoarea jucătorului, atunci evenimentul se consumă și algoritmul se oprește aici, în orice alt caz mergem mai departe.
- 5) Verificăm dacă piesa selectată era rege, dacă da verificăm dacă a fost făcută rocada și dacă da, atunci ce tip de rocada, unde (1 - rocada scurtă, 2 - rocada lungă, 0 - nici o rocada). Tipul rocadei se salvează într-o variabilă specială `castlingType` de tip short. Dacă tipul rocadei este 1 sau 2, atunci turnul se mișcă conform regulilor de șah, în caz contrar algoritm merge mai departe.

P.S. Rocada se face doar în caz când între turnul și rege nu există nici o piesa, ambele piese sunt de aceeași culoare și ei nu erau mișcate niciodată până la momentul respectiv.

- 6) Se verifică dacă piesa selectată este un pion, în caz afirmativ verificăm dacă am ajuns la sfârșitul opus a tablei, în caz afirmativ arătăm pe ecran fereastră modală pentru selectarea piesei noi. În moment ce fereastră modală era apelată algoritmul se oprește și așteaptă răspunsul de la fereastră. După răspuns pe poziția *inițială* se pune piesa selectată în fereastră modală și algoritmul merge mai departe
- 7) De aici codul rulează în orice alt caz și după subpunctul 5 și 6. Noi punem în matricea, care păstrează conținutul tablei mișcăm piesa de la poziția inițială spre poziția finală. Dacă subpunctul 6 era făcut, atunci în variabilă specială `move` initializăm un nou obiect `new Move(x, y, x1, y1, <numele piesei selectate>)`, în caz contrar initializăm un nou obiect cu constructor alternativ `new Move(x, y, x1, y1, castlingType)`, care pe default are valoarea de 0 (nu este nici o rocada).
- 8) Ștergem conținutul celulelor pe partea vizuală și în celulă finală punem piesa mișcată.
- 9) Pe label, care ne spune informații importante despre starea jocului se pune mesaj, care spune jucătorului să termine mișcarea lui.

- 10) Se produce sunet de mișcarea a piesei
- 11) Se ia dreptul jucătorului de a mai mișca piese
- 12) Variabila care păstra piesa selectată din fereastră modală se face null, pentru a nu produce erori în mișcări din viitor.
- 13) La momentul dat programul așteaptă până când jucătorul apese pe unul din butoane disponibile: “End Turn”, “Check”, “Checkmate”, “Surrender”. Apăsarea pe primii 3 butoane nu este posibilă până când jucătorul nu a mișcat piesa. Aceasta se regulează cu ajutorul variabilei move, care păstrează mișcarea curentă. Dacă variabila este nulă, atunci butoanele sunt blocate. Dacă erau apasate una din primii 3 butoane, programul trimite spre server mesaj, care conține informații despre mișcarea efectuată. ([“Check” | “Checkmate”][<tipul de castling> | <denumirea piesei selectate>] x y x1 y1). După ce setează variabilă move pe null, pentru a nu cauza probleme că jucătorul poate să finalizeze mișcarea înainte de a face una. Și label pentru informații afișează mesaj, că trebuie de așteptat mișcarea oponentului.

P.S. codificarea și ce înseamnă fiecare simbol din mesaj poate fi găsită în partea “Definiții, Acronime și Abrevieri”

- 14) Dacă era apăsat butonul “Surrender”, care poate fi apăsat oricând, chiar și în timpul mișcării oponentului, pe label pentru informații este afișat “Defeat!”, și jucătorul se consideră că a pierdut. Jucătorul transmite mesajul “surrender” către server și închide conexiunea. Jucătorul nu mai poate face mișcări, chiar dacă era mișcarea lui. Toate butoanele se blochează și apare buton nou “Close Game”, care închide fereastra de joc și întoarce pe jucător în meniul principal, unde el poate începe un joc nou, sau să închidă aplicația.

Algoritmul asemănător, dar puțin diferit nu ca implementarea, dar ca utilizarea se mai întâlnește în partea de client a aplicației, care obține mesaje de pe server, pe care am s-o descriu într-o funcționalitate aparte.

2.2.2 Fluxuri alternative

2.2.2.1 < Primul flux alternativ >

Programul poate transmite între client server nu numai șiruri de caractere, dar și obiecte, de aici vine ideea, că este posibil de a crea o clasă “Răspuns”, care va avea ca proprietăți câteva câmpuri de tip boolean. În felul acesta ar fi fost mult mai simplu de a trimite și de a prelucra date transmise.

2.2.3 Pre-condiții

Pentru ca funcționalitate să funcționeze, jucătorul trebuie să fie conectat în joc, și trebuie să fie mișcarea lui. Server trebuie să fie pornit.

2.2.4 Post-condiții

În rezultatul rulării funcționalității pe label de informații va fi scris “Defeat!”, “Victory!” sau “Waiting for another player’s move...”, “It is your turn” și piesa va fi mișcată

3 Funcționalitate. Obținerea mesajelor de pe server de către jucător

3.1 Descriere

În proiect există clasă Player, care reprezintă partea de client. Ea obține mesaje de pe server, le procesează și face schimbările, care trebuie să fie făcute conform mesajului obținut.

P.S. Informații suplimentare despre codificarea mesajelor poate fi găsită în partea “Definiții, Acronime și Abrevieri”

3.2 Fluxul de evenimente

3.2.1 Fluxul de bază

- 1) Programul primește răspunsul de pe server și îl afișează pe consolă, pentru a ușura debugging
- 2) Intră într-un switch, care verifică dacă răspuns este “disconnected”, “surrender”, “WHITE”, “BLACK” sau “connected”. Dacă este “disconnected”, atunci jucător tot se deconectează de pe server. Mesajul acesta înseamnă că jucător opus a închis jocul. Dacă este “surrender”, atunci jucătorul tot se deconectează de la server și afișează în label pentru informații “Victory!”. Dacă este “BLACK”, “WHITE”, sau “connected” atunci dacă jucător nu are culoare setată, setează culoarea transmisă. După ce (de-aici începe în cazul “connected”) dacă culoarea jucătorului este alb, atunci este mișcarea lui, ce scrie pe label de informații.
- 3) În orice alt caz mesajul se rupe în locuri de spații. Se creează variabilă i, care specifică de pe ce poziția se începe coordonatele în mesaj obținut. Programul verifică primii doi membrii al vectorului de stringuri obținut, deoarece poate fi situația, când facem schimbarea de piesă și apasam pe butonul “Check” în situația aceasta în mesajul obținut până la coordonatele vor fi 2 cuvinte importante pentru reproducerea corectă a mișcării. Dacă este “Check”, sau “Checkmate”, atunci i se incrementează, dacă este “shortCastling” atunci setează variabilă castlingType pe 1, dacă este “longCastling” atunci setează variabilă castlingType pe 2, apoi verifică toate denumiri a pieselor posibile, dacă este cel puțin una din ele, atunci variabilă specială selectedPiece se inițializează cu constructor respectiv denumirii obținute.
- 4) Se pune în x y coordonatele de start, și în coordonatele x1 y1 coordonatele finale unde s-a mutat piesa. Știind coordonatele se obțin celule din tablou, care participă în mișcarea.
- 5) Se prelucrează rocada, dacă variabila castlingType nu este nulă. Și se prelucrează schimbarea piesei, dacă variabilă selectedPiece nu este nulă.
- 6) Se face mișcarea pieselor în matricea, care conține starea tablei.
- 7) Se mișcă piese pe partea vizuală a aplicației.
- 8) Dacă primul cuvânt era “Check” sau “Checkmate”, atunci în label pentru informații se scrie “Check” sau “Checkmate” respectiv. În caz contrar se scrie “It is your turn!”
- 9) Se pune variabilă isYourTurn pe true.

3.2.2 Fluxuri alternative

3.2.2.1 < Primul flux alternativ >

Programul poate transmite între client server nu numai șiruri de caractere, dar și obiecte, de aici vine ideea, că este posibil de a crea o clasă “Răspuns”, care va avea ca proprietăți câteva câmpuri de tip boolean. În felul acesta ar fi fost mult mai simplu de a trimite și de a prelucra date transmise.

3.2.3 Pre-condiții

Jucătorul trebuie să fie conectat pe rețea locală cu jucător, care va începe jocul. Server trebuie să fie pornit.

3.2.4 Post-condiții

În rezultatul rulării funcționalității pe label de informații va fi scris “Defeat!”, “Victory!”, “It is your turn”, “Check” sau “Checkmate” și piesa(-e) va fi mișcată.

4 Funcționalitate. Server

4.1 Descriere

Server răspunde pentru transmiterea informațiilor între jucători și să le conecteze împreună.
Server este o aplicație aparte

4.2 Fluxul de evenimente

4.2.1 Fluxul de bază

- 1) El așteaptă până când cineva încearcă să se conecteze. Când aceasta se întâmplă el verifică dacă la joc sunt conectați două jucători, dacă nu, atunci el conectează pe aceste cineva în joc și scrie în consolă “A player was connected”.
- 2) Pornește un cod în paralel, care va aștepta un mesaj de la jucător, în caz dacă jucător o să iasă până ca jocul să se înceapă
- 3) Când două jucători sunt conectate jocul se va începe.
- 4) Dacă server primește mesaj de la un jucător el îl transmite altui jucător, după ce verifică dacă mesajul este “surrender” sau “disconnected”, în caz afirmativ el deconectează amândouă jucători și termină jocul. La aceste moment server este gata pentru a conecta noi jucători.

4.2.2 Fluxuri alternative

4.2.2.1 < Primul flux alternativ >

Programul poate transmite între client server nu numai șiruri de caractere, dar și obiecte, de aici vine ideea, că este posibil de a crea o clasă “Răspuns”, care va avea ca proprietăți câteva câmpuri de tip boolean. În felul acesta ar fi fost mult mai simplu de a trimite și de a prelucra date transmise.

4.2.3 Pre-condiții

Jucători trebuie să se afle într-o rețea locală. Server trebuie să fie pornit.

4.2.4 Post-condiții

Dacă totul este corect, jucători voi vedea tabla cu piese rotită cu piesele lor spre ei. În consolă de către programator se poate de văzut mesaj “connected” sau “BLACK” | “WHITE”. Jucători voi vedea în label de informații mesaje “It is your turn!” sau “Waiting for another player’s move...”

5 Funcționalitate. Start de joc

5.1 *Descriere*

Inițializarea jocului. Este important pentru ca să precizez niște precondiții care este important de menționat

5.1.1 Pre-condiții

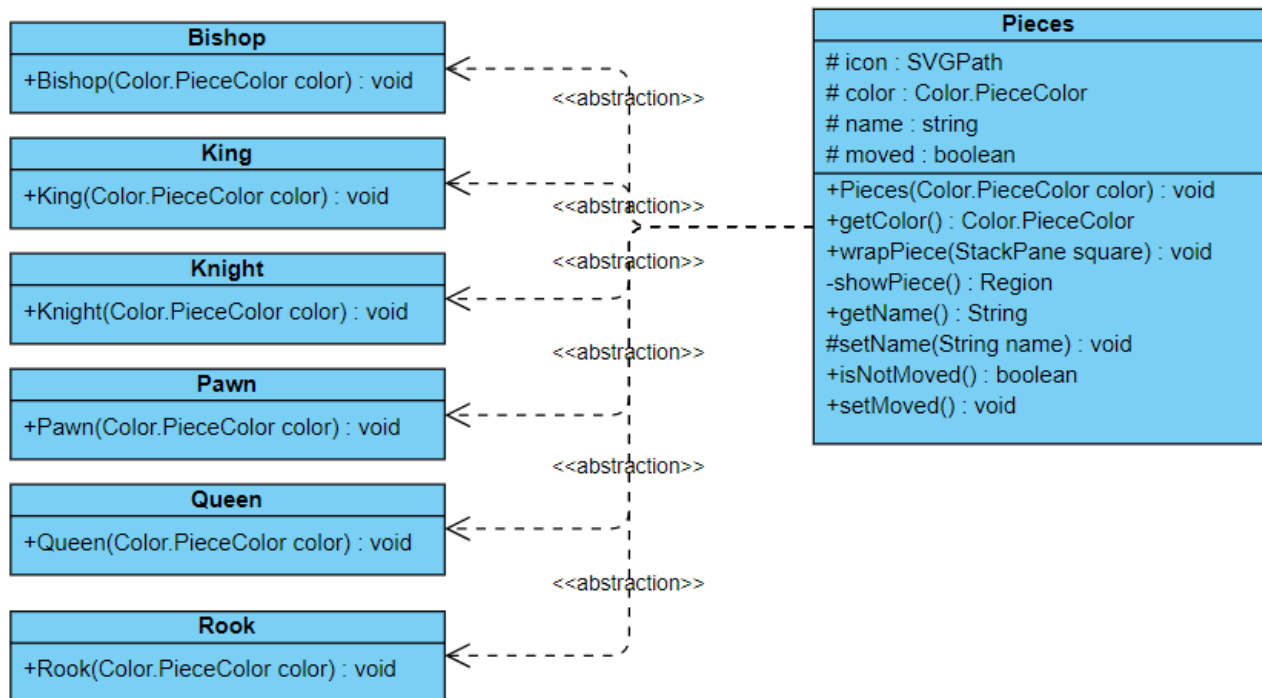
Jucător care inițializează jocul trebuie să apese pe butonul “Start Game”, și să selecteze culoarea dorită. Jucător, care vrea să se conecteze trebuie să apese pe butonul “Find Game”. Aceasta este rezolvarea temporală pentru a implementa în viitor funcționalitatea de menținerea mai multor jocuri simultan.

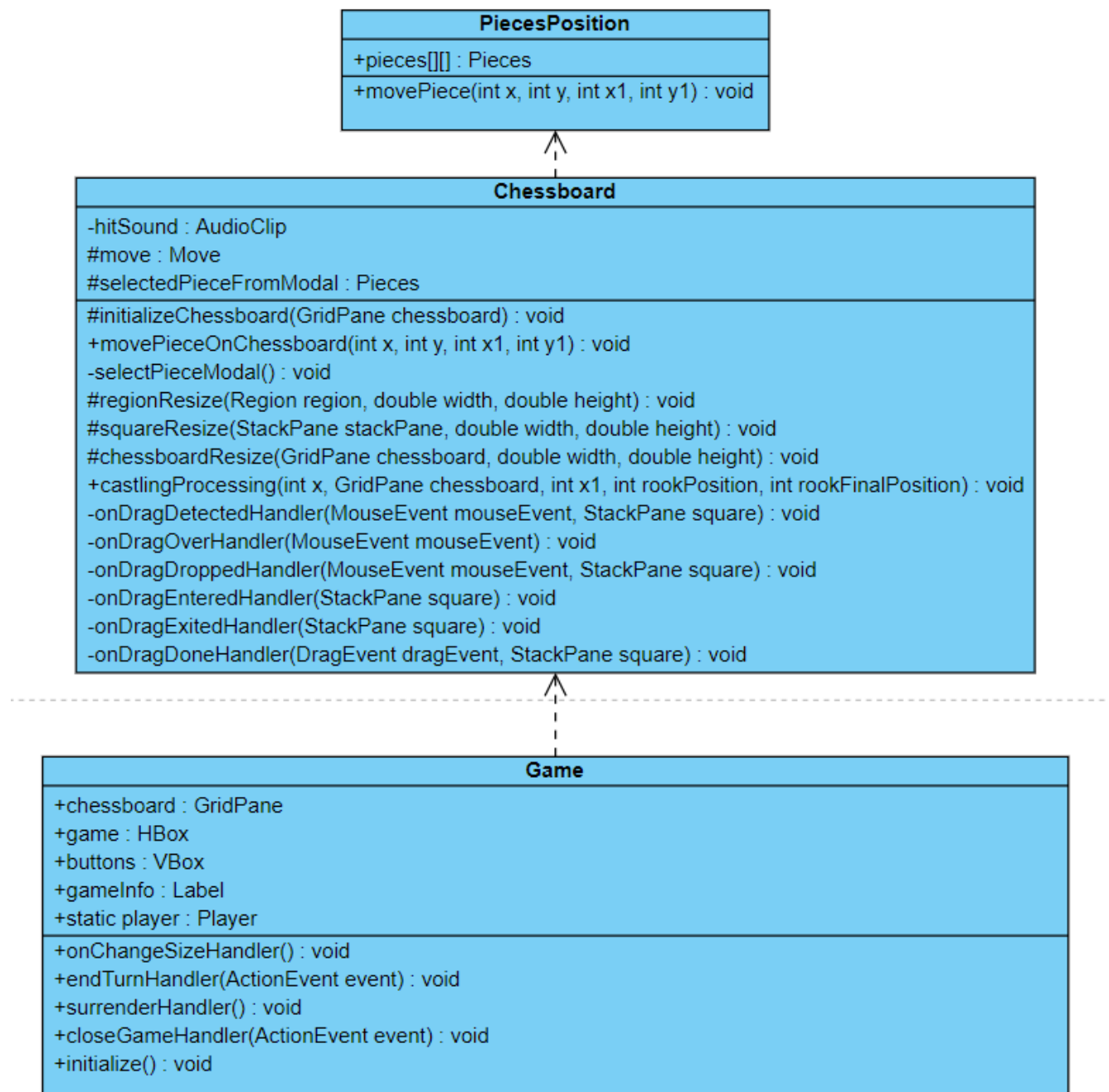
5.1.2 Post-condiții

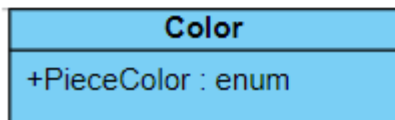
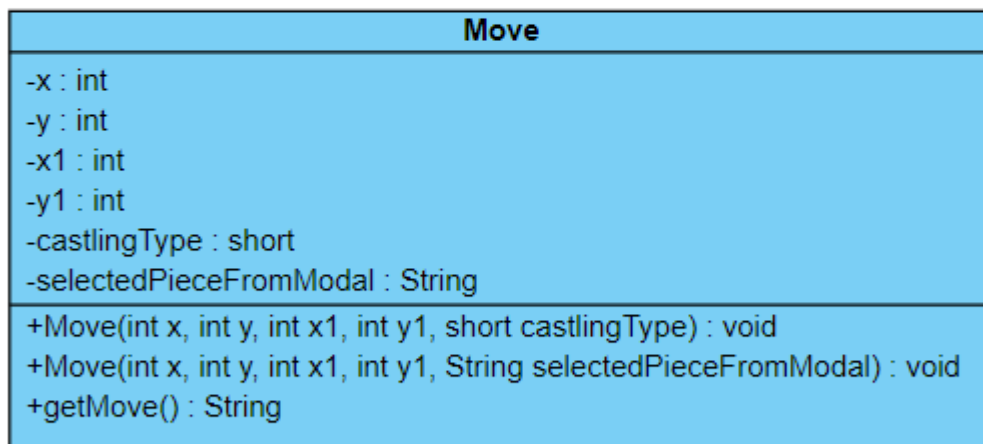
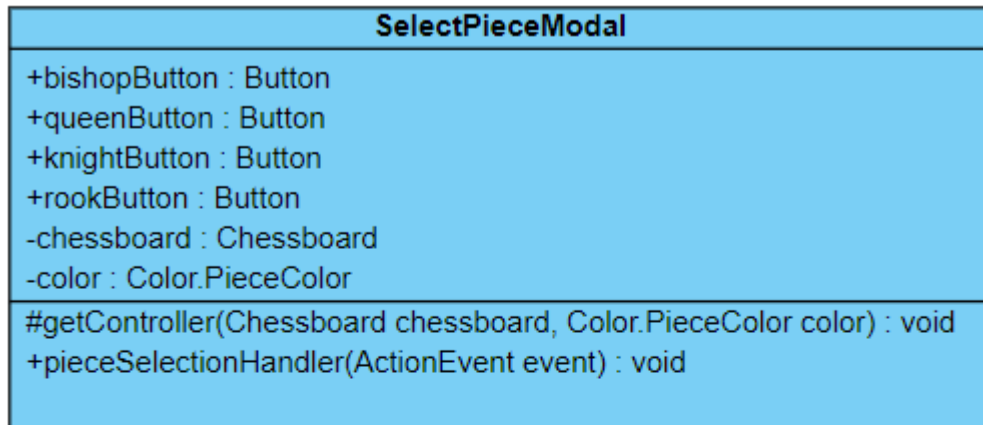
Jucătorul, care a început jocul va vedea tabla cu piese de culoarea selectată rotită spre lui și în label pentru informații va fi afișat “Waiting for another player...”

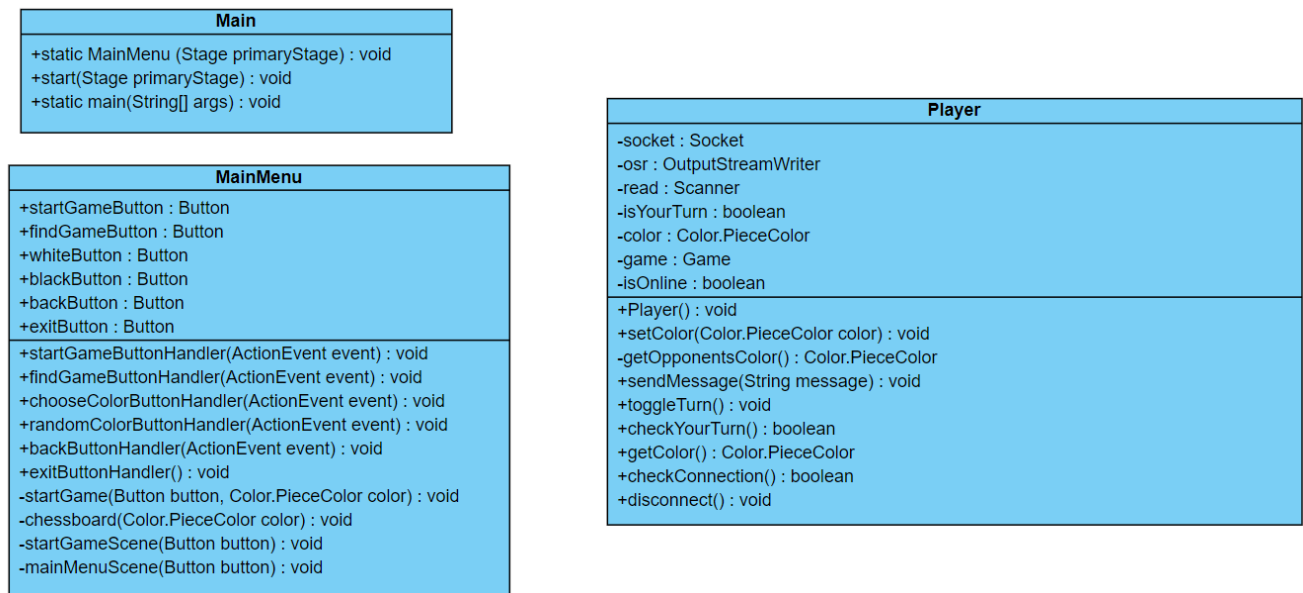
6 Implementare

6.1 Diagrama de clase (Joc)

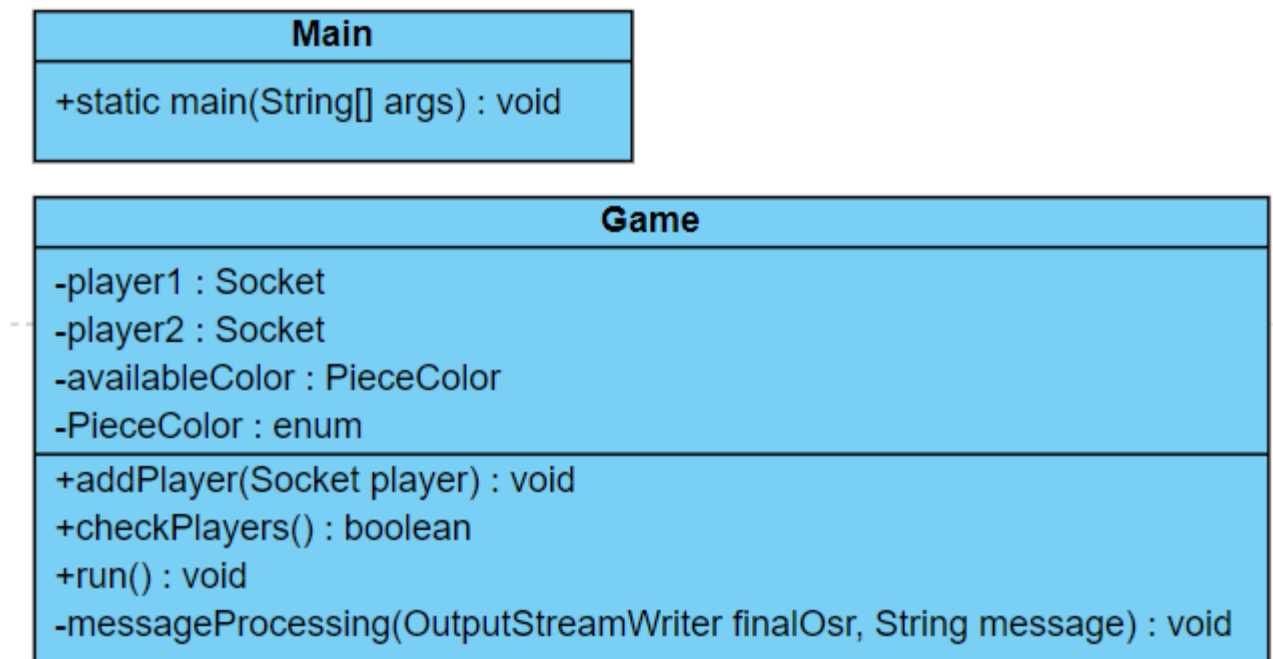






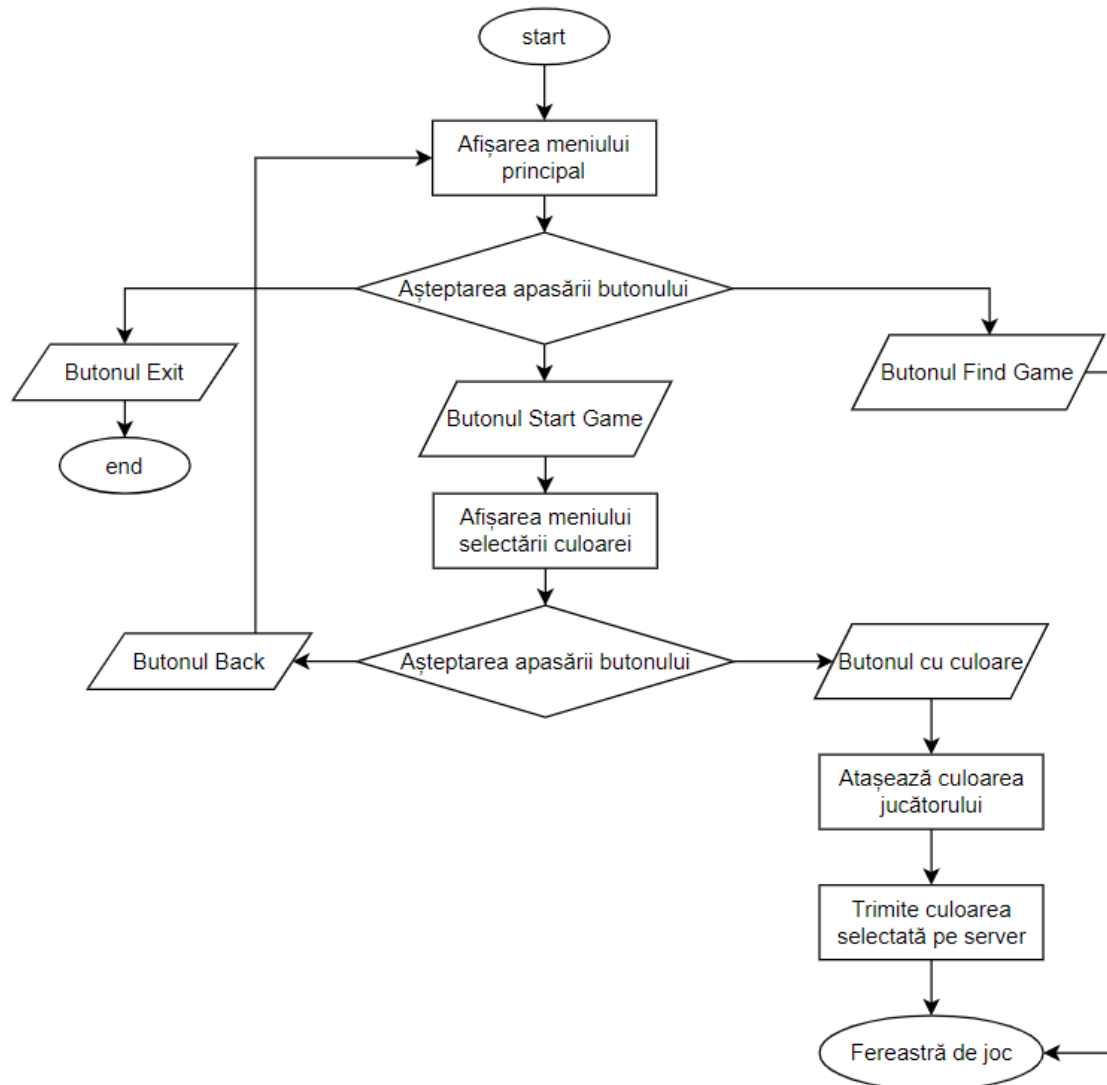


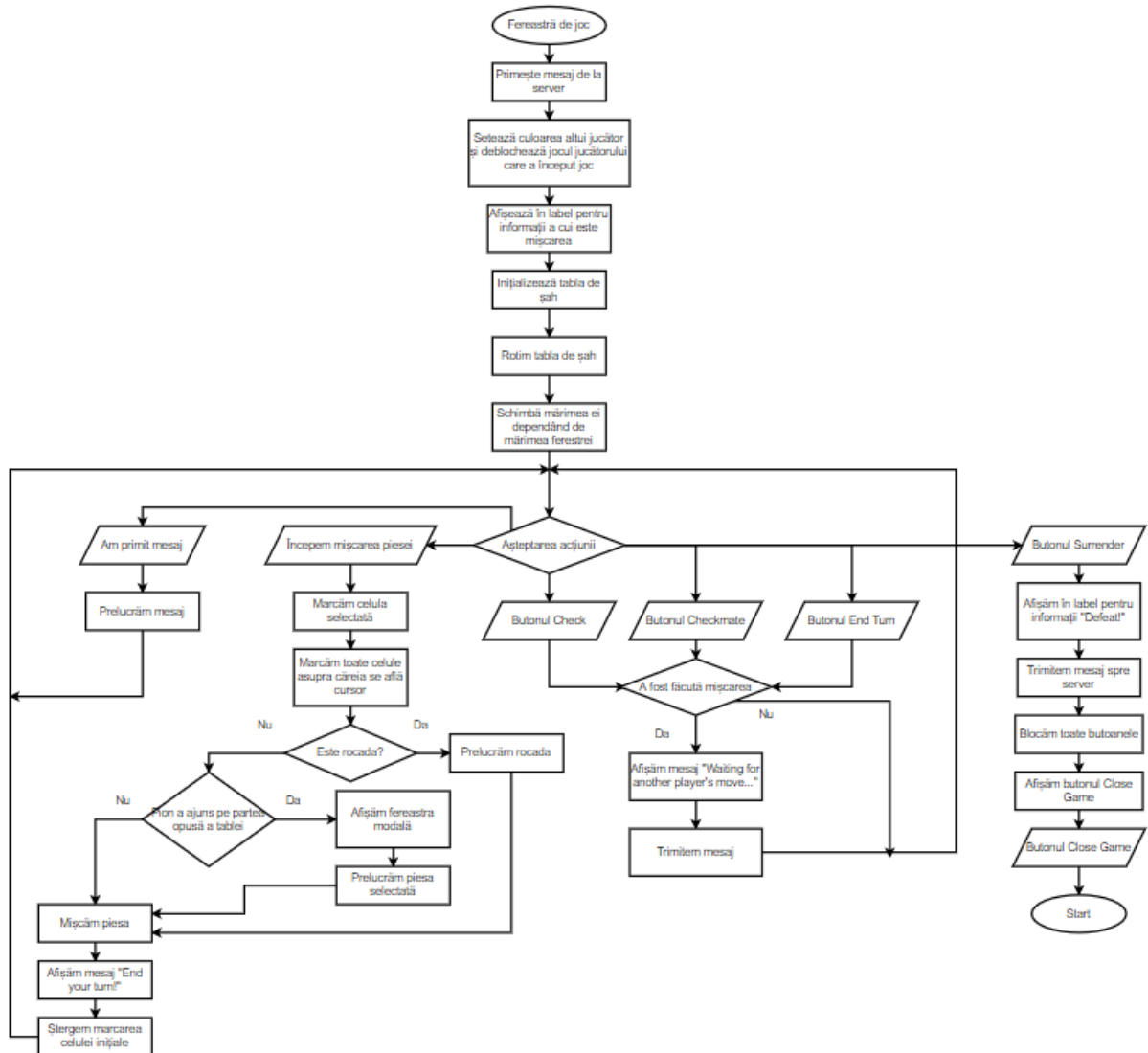
6.2 Diagrama de clase (Server)



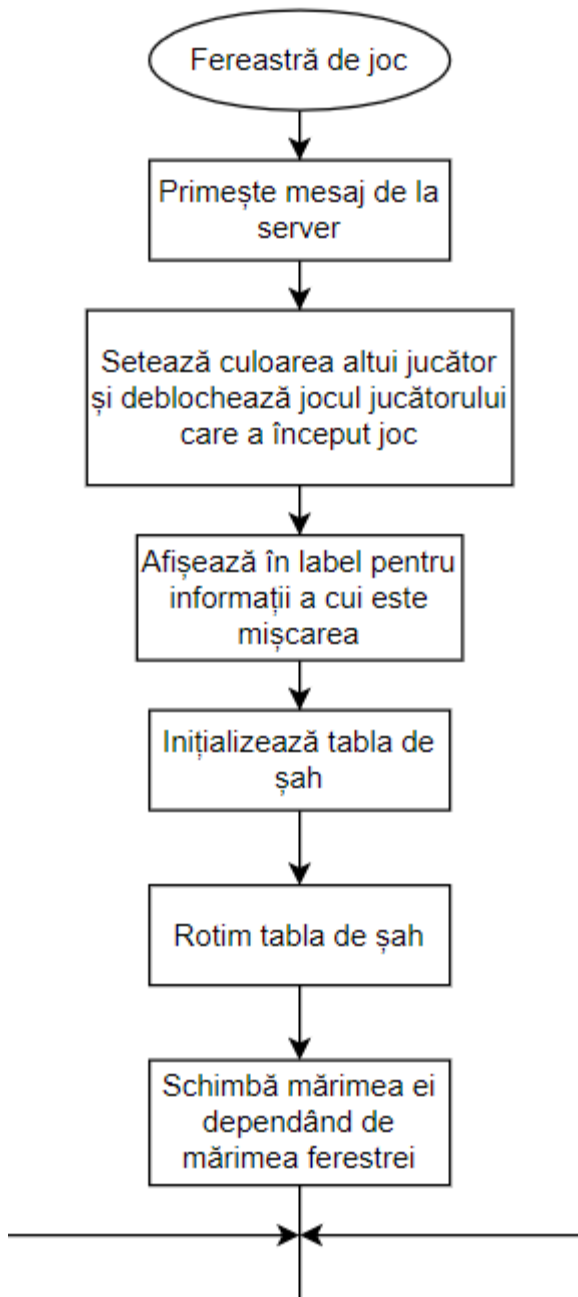
6.3 Descriere detaliată

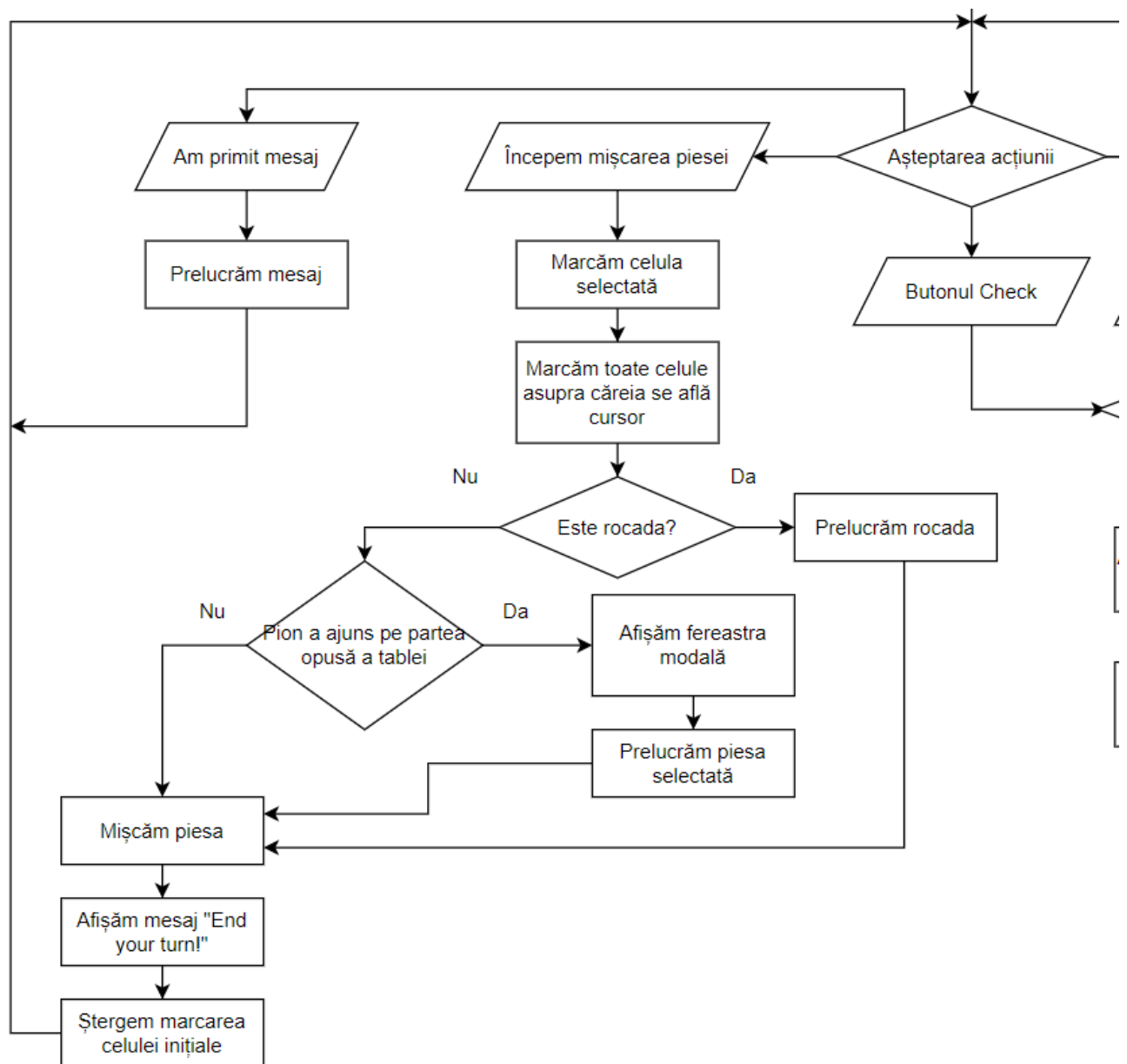
Meniul principal:

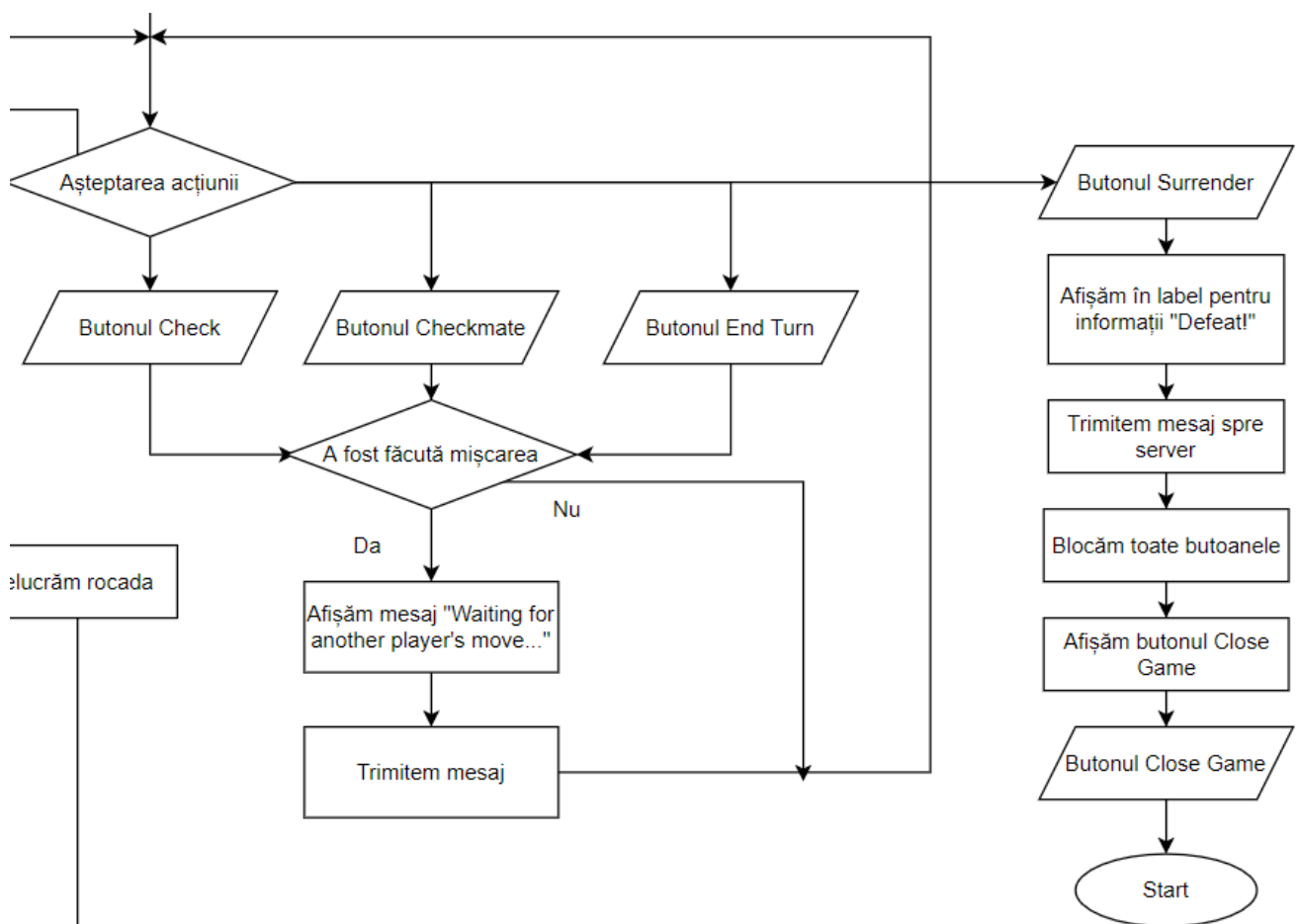


Fereastră de joc:

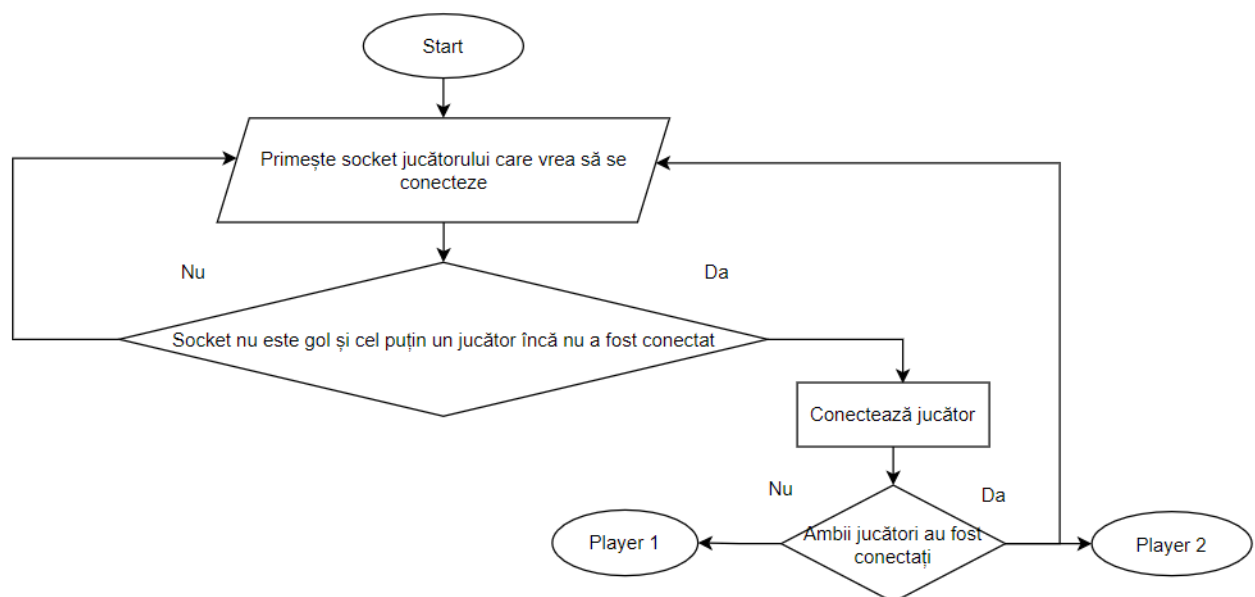
Mai apropiat:

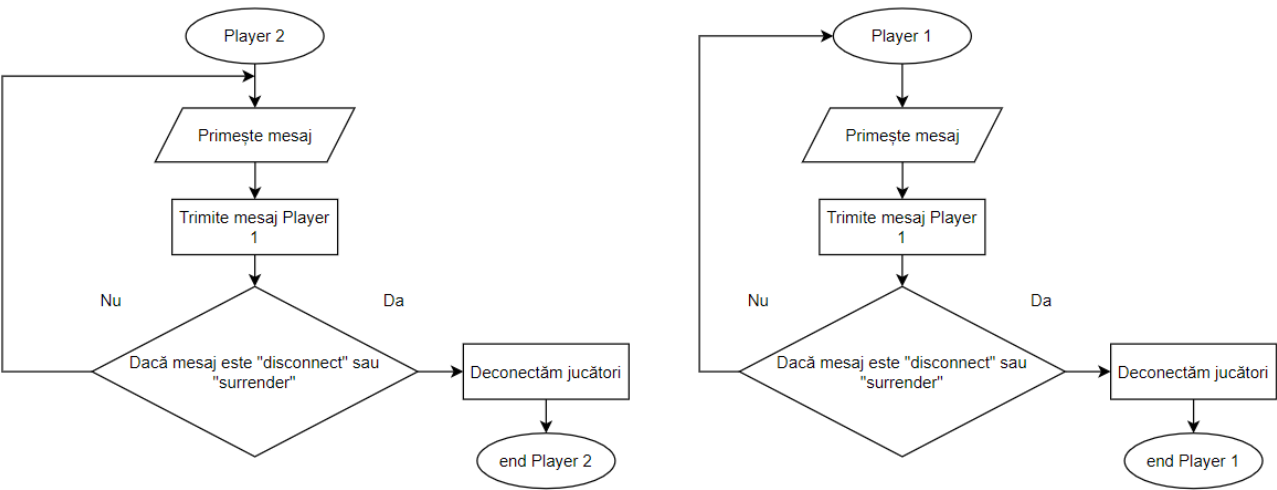






Server:





7 Bibliografie

- [1] monograff76 - Drag-and-Drop Events
<https://monograff76.wordpress.com/2015/02/20/developing-a-drag-and-drop-ui-in-javafx-part-3/>
- [2] tutorialspoint - JavaFX - 2D Shaped SVG
https://www.tutorialspoint.com/javafx/2dshapes_svgpath.htm
- [3] tutorials.jenkov - JavaFX Stage
<http://tutorials.jenkov.com/javafx/stage.html>
- [4] tutorials.jenkov - JavaFX GridPane
<http://tutorials.jenkov.com/javafx/gridpane.html>
- [5] Oracle docs - Audio Clip
<https://docs.oracle.com/javafx/2/api/javafx/scene/media/AudioClip.html>
- [6] codeflow - Руководство по сокетам Java
<https://www.codeflow.site/ru/article/a-guide-to-java-sockets>
- [7] coderlessons - JavaFX, сокеты и потоки: извлеченные уроки
<https://coderlessons.com/articles/java/javafx-sokety-i-potoki-izvlechennye-uroki>
- [8] medium - Server Client Chat Application with Java Socket Programming
<https://bundet.medium.com/server-client-chat-application-with-java-socket-programming-70374ac75448>
- [9] callicoder - Styling JavaFX applications using CSS
<https://www.callicoder.com/javafx-css-tutorial>
- [10] jetbrains - Create a new JavaFX project
<https://www.jetbrains.com/help/idea/javafx.html>
- [11] jetbrains - Configure JavaFX Scene Builder
<https://www.jetbrains.com/help/idea/opening-fxml-files-in-javafx-scene-builder.html>
- [12] devto - JavaFX: 3 Ways of Passing Information Between Scenes!
<https://dev.to/devtony101/javafx-3-ways-of-passing-information-between-scenes-1bm8>
- [13] reddit - Problem with updating GUI (javafx)
https://www.reddit.com/r/javahelp/comments/7qvqau/problem_with Updating_gui_javafx/