

Nu te supăra frate

14 Ianuarie 2020

Îndrumător:

dr. ing. Daniel Morariu

Student:

Țoghe Răzvan-Constantin

(223/ 2)

Istoric Versiuni

Data	Versiune	Descriere	Autor
30-Oct	0.0.100	Creare interfata grafica (cercuri pe ecran)	Toghe Razvan
	0.0.101	Creare piese jucatori, butoane, zarul virtual	Halmaghi Denisa
30-Oct	0.0.102	Modificarea cu un nume adecvat fiecărei componente din proiect	Toghe Razvan
31-Oct	0.0.110	Blocare margini fereastra pentru a nu redimensiona jocul	Toghe Razvan
	0.0.111	Editare dimensiuni piese jucatori	Toghe Razvan
	0.0.112	Editare si calculare dimensiuni, distanta dintre cercuri	Toghe Razvan
01-Nov	0.0.113	Crearea, afisarea și mutarea primelor obiecte dinamice	Toghe Razvan
02-Nov	0.0.200	Crearea unei ferestre pentru alegere numarul de jucatori	Toghe Razvan
	0.0.201	Crearea clasei Pozitie și derivata ei Galben	Toghe Razvan
03-Nov	0.0.210	Crearea și implementarea constructorilor pentru Pozitie și Galben	Toghe Razvan
	0.0.211	Implementarea unui vector de pozitii prin care sa mutam	Toghe Razvan
	0.0.212	Mutarea tuturor obiectelor globale în clasa forme	Toghe Razvan
	0.0.213	Implementarea unor vectori de pozitii pentru locatiile de inceput și final de joc	Toghe Razvan
04-Nov	0.0.220	Încercarea de a muta vectorul de pozitii ca membru al unei clase	Toghe Razvan
	0.0.221	Crearea unui alt vector de pozitii în clasa derivata pentru a nu mai utiliza vectorul global	Toghe Razvan
	0.0.222	Implementarea de metode pentru mutarea și utilizarea noului vector de pozitii	Toghe Razvan
05-Nov	0.0.223	Adaugarea de culori, font-uri componentelor	Toghe Razvan
08-Nov	0.0.300	Renuntare la fereastra pentru alegere a numarului de jucatori	Toghe Razvan
	0.0.301	Adaugarea de butoane pentru ca fiecare jucator sa aleaga ce culoare vrea	Toghe Razvan
08-Nov	0.0.302	Editare metoda de a muta piesele	Borland
	0.0.303	Extindere a vectorului de pozitii pentru toate piesele	Toghe Razvan
09-Nov	0.0.310	Implementarea primelor conditii de joc pentru a iesi din baza	Toghe Razvan
	0.0.311	Implementare polimorfism pentru metodele Aseaza și Joaca	Toghe Razvan
10-Nov	0.0.320	Tranzitia evenimentelor de click pe piesa în metoda clasei de baza	Toghe Razvan
	0.0.321	Încercare esuata de a muta clasele în unit-uri independente	Toghe Razvan
23-Nov	0.1.000	Mutarea claselor în unit-uri independente	Toghe Razvan
	0.1.100	Transmiterea ca parametrii adresele variabilele "globale" pentru majoritatea metodelor	Toghe Razvan
	0.1.110	Rezolvarea de bug-uri & cleancode	Toghe Razvan
24-Nov	0.1.120	Crearea unei metode iaParametrii care ia adresele tuturor variabilelor necesare jocului	Toghe Razvan
	0.1.121	Modificarea metodei Joaca pentru a primi toate adresele variabilelor necesare	Toghe Razvan
	0.1.122	Modificarea restului de metode pentru a se integra în Joaca	Toghe Razvan
04-Dec	0.1.130	Modificarea metodei iaParametrii pentru a primi pozitiile pieselor aliate	Toghe Razvan
	0.1.140	Crearea unei metode verificaPozitii_Scrie pentru a afisa un numar pe piese daca cele aliate se suprapun	Toghe Razvan
	0.1.141	Crearea unui memo pentru Test și debugging pozitii în timpul mutarilor	Toghe Razvan

	0.1.142	Editarea metodelor de mutare, verificare drum și ieșire din casa	Toghe Razvan
05-Dec	0.1.143	Modificarea metodei verificaPozitii_Scrie pentru a acoperii toate cazurile	Toghe Razvan
	0.1.144	Adaugarea unui timer care aplica metoda verificaPozitii_Scrie la fiecare 0.1s	Toghe Razvan
13-Dec	0.1.150	Rezolvarea de bug-uri & cleancode	Toghe Razvan
14-Dec	0.1.160	Incercare de implementare retea esuata	Toghe Razvan
18-Dec	0.1.170	Continuare dezvoltare de functionalitati	Toghe Razvan
	0.1.180	Implementarea unei metode care la miscarea mouse-ului deasupra piesei sa se facă highlight pe cercul unde urmeaza sa se mute	Daniel Morariu
	0.1.181	Modificarea metodei iaParametrii pentru a primii adresele a 2 vectori TShape	Toghe Razvan
	0.1.190	Rezolvarea de bug-uri & cleancode	Toghe Razvan
	0.1.191	Adaugare iconita joc	Toghe Razvan
19-Dec	0.1.200	Optimizarea codului pentru a trimite doar adresa cercului caruia sa se facă highlight, nu tot vectorul	Toghe Razvan
	0.1.210	Modificarea metodelor de joc pentru a se deplasa câte o pozitie pana la pozitia finala - Bug, raman urme ale piesei la mutare	Toghe Razvan
	0.1.220	Adaugarea de sunete pentru ieșire din casa,mutare,intrare final (FL Studio 20)	Toghe Razvan
22-Dec	0.2.000	Adaugare de animatie la zar 2 (Blender 2.8 + Photoshop CC 2019)	Toghe Razvan
27-Dec	0.3.000	Adaugare restul animatii pentru zar + Ajustare + implementare cod	Toghe Razvan
28-Dec	0.3.100	Micsorare animatii zar - bug, de cum se intra pe timmer gif-ul se face vizibil instant daca am mai dat o data acea valoare cu zarul	Toghe Razvan
	0.3.200	Cleancode	Toghe Razvan
	0.4.000	Adaugare clasa pentru piesele Verzi	Toghe Razvan
	0.4.100	Modificarea metodelor de joc pentru a se potrivi pieselor Verzi	Toghe Razvan
	0.5.000	Adaugare clasa pentru piesele Rosii	Toghe Razvan
	0.5.100	Modificarea metodelor de joc pentru a se potrivi pieselor Rosii	Toghe Razvan
	0.6.000	Adaugare clasa pentru piesele Albastre	Toghe Razvan
	0.6.100	Modificarea metodelor de joc pentru a se potrivi pieselor Albastre	Toghe Razvan
	0.6.200	Rezolvarea de bug-uri & cleancode	Toghe Razvan
30-Dec	0.7.000	Implementarea turei fiecarui jucator (pentru 1 pana la 4 jucatori)	Toghe Razvan
01-Jan	0.8.000	Implementare metode pentru a baga în casa inamicii, a verifica casele și disponibilitatea jucatorului	Toghe Razvan
03-Jan	0.9.000	Inceput de rețea	Toghe Razvan
10-Jan	0.9.010	Creare fereastra Lobby	Toghe Razvan
	0.9.020	Bug fix dublare piese dupa transmitere pe retea	Toghe Razvan
	0.9.030	Modificare cod server pentru a transmite datele pentru lobby	Toghe Razvan
11-Jan	0.9.100	Finalizare lobby	Toghe Razvan
12-Jan	1.0.000	Finalizare retea pentru mutari, ieșire din casa, intrare final, zar=6	Toghe Razvan
13-Jan	1.0.010	Cleancode	Toghe Razvan

Cuprins

ISTORIC VERSIUNI	2
CUPRINS	4
1 SPECIFICAREA CERINȚELOR SOFTWARE	5
1.1 Introducere	5
1.1.1 Obiective	5
1.1.2 Definiții, Acronime și Abrevieri	6
1.1.3 Tehnologiile utilizate	6
1.2 Cerințe specifice.....	6
2 FUNCȚIONALITATE.....	7
2.1 Descriere.....	7
2.2 Fluxul de evenimente	7
2.2.1 Fluxul de bază	7
2.2.2 Fluxuri alternative	9
2.2.3 Pre-condiții.....	9
2.2.4 Post-condiții	10
3 IMPLEMENTARE	11
3.1 Diagrama de clase.....	11
3.2 Descriere detaliată.....	112
4 BIBLIOGRAFIE	13

1 Specificarea cerințelor software

1.1 Introducere

Jocul pornește cu imaginea tablei și un buton de conectare, care, odată apăsă, ne putem alege dacă vrem să jucăm cu mai mulți în rețea sau pe un singur PC. Dacă vrem să jucăm în rețea, introducem adresa IP a calculatorului, un nume de jucător și eventual, dacă am schimbat port-ul, introducem adresa lui, implicit este portul 2001. Apăsăm pe conectare și o să intrăm într-o cameră de așteptare în care jucătorii își aleg cu ce culoare vor să joace, după care bifează căsuța „Gata” din dreptul lor. Când toți jucătorii au bifat că sunt gata, jocul pornește după numărătoarea inversă care durează 5 secunde.

Dacă am selectat jocul pe rețea, piesele și rândul fiecăruia se v-or face automat în funcție de opțiunile alese de jucători în camera de așteptare, dacă am selectat jocul fără rețea, fiecare jucător trebuie să dea click pe buton-ul pentru culoarea cu care vrea să joace.

După ce toți jucătorii au pus piesele pe tablă, jocul începe apăsând pe zar, v-a urma apoi o animație a zarului, după care jucătorul al cărui turn este, poate da click pe piesa pe care vrea să o mute.

Dacă un jucător are mai multe piese în aceeași poziție, în interiorul piesei se v-a afișa un mesaj corespunzător cu numărul de piese aflate pe aceeași poziție, inclusiv dacă aceasta se află în casă sau la final.

În momentul în care jucătorul mută mouse-ul deasupra piesei pe care vrea să o mute, cercul în care piesa se v-a deplasa o să se coloreze în albastru pentru ca jucătorului să îi fie mai ușor de făcut o strategie și să nu mai calculeze unde v-a ateriza piesa.

Dacă piesa unui jucător ajunge în poziție finală, se verifică dacă un alt jucător are vreo piesă în acel loc, dacă se dovedește a avea, acea piesă v-a ajunge înapoi în casă.

După ce piesa unui jucător a străbătut toată tabla, piesa poate să intre la final.

1.1.1 Obiective

- Disponibilitate până la 4 jucători.
- Piese pentru toți jucătorii.
- Case și final pentru toți jucătorii.
- Tablă de joc.
- Zar.
- Posibilitate de a muta piesele pe tablă.
- Posibilitate de a vedea unde o să ajungă piesa până să o mutăm.
- Condiții pentru mutări.
- Tură pentru jucători.
- Sunete și animații.
- Rețea.
- Cameră de așteptare.

- Chat.
- Mesaje interactive pentru diverse acțiuni.
- Un panou cu realizările și acțiunile efectuate până la finalul meciului.
- Opțiuni de a schimba culoarea de fundal, muzică.
- Muzică de fundal.

Obiectivele marcate cu “ – „ au fost finalizate

1.1.2 Definiții, Acronime și Abrevieri

- **Casă** – Cele 4 poziții marcate cu culoare potrivită jucătorului în care sunt așezate piesele la începutul jocului.
- **Final** – Cele 4 poziții în care jucătorul își v-a pune piesele după ce a străbătut toată tabla.
- **Tablă** – 40 de poziții pe care jucătorii se deplasează, tabla fiecărui jucător începe de la poziția colorată în culoarea jucătorului, se continuă în sensul acelor de ceasornic până ajunge la poziția de dinaintea începerii.
- **Poziție finală** – Poziția la care piesa s-a deplasat pe tablă.
- **Tură** – Fiecare jucător trebuie să își aștepte rândul pentru a putea da cu zarul și a se deplasa, ordinea se stabilește în funcție de cine s-a conectat mai repede la rețea, sau de alegerile făcute în modul singleplayer.
- **Cerc de tablă** – Cele 40 de cercuri ale tablei.
- **Cerc de final** – Cele 4 cercuri ale finalului.
- **Cerc de bază** – Cele 4 cercuri ale bazei.
- **Relativ la observator** – Jucătorii văd obiectele așa cum sunt văzute pe ecran.
- **Observator** – Cei care urmăresc jocul din fața ecranului.
- **Identitatea** – Piesa se poate identifica prin 2 membrii, un șir de caractere, care reține culoarea, și o valoare care reține numărul piesei, fiecare jucător are 4 piese, iar culorile existente sunt: Galben, Verde, Rosu, Albastru.
- **Aux0** – În general folosit ca variabilă pentru butonul de zar, dar ca și o poziție de tip TPoint.

1.1.3 Tehnologiile utilizate

- Componentă externă ActiveX numită AniGif.
- Blender pentru a crea cadrele animației la zar
- Adobe Photoshop Creative Cloud 2019 pentru a edita cadrele și a le integra într-un GIF
- FL Studio 20 pentru adăugarea de efecte sonore
- Audacity pentru modificarea perioadei și amplitudinii efectelor sonore
- Draw.io pentru schemele de la finalul documentului

1.2 Cerințe specifice

- Disponibilitate până la 4 jucători – s-au adaptat codurile la o variabilă numită `numarJucatori` pentru a se putea verifica tura fiecăruia, a afișa text în interiorul piesei, a afișa numele jucătorilor și piesele.
- Afișarea pieselor pe tabla de joc pentru jucători.
- Crearea de locuri speciale pentru jucători – bază și final.
- Crearea tablei de joc și a pozițiilor aferente fiecărei locații de pe tablă, casă și final. Fiecare piesă știe pe unde se poate plimba.
- Crearea zarului, asignarea sa a unei valori din intervalu [1:6] și pornirea unei animații specifice valorii.
- Transmiterea parametrilor către piese.
- Mutarea pieselor pe tablă.
- Highlight pe poziția unde se v-a muta zarul.
- Condiții pentru mutări – să fie rândul tău, să dai mai întâi cu zarul, dacă ai dat cu zarul 6, poți să ieși din casă și mai poți da din nou cu zarul, dacă ai ajuns la final poți da din nou cu zarul, dacă cineva mută o piesă în poziție finală peste o piesă de-a ta, piesa ta intră în primul loc liber din casă.
- Tură pentru jucători.
- Sunete pentru mutarea piesei, pentru ieșirea din casă și pentru intrarea în final.
- Codificarea realizată pentru mesajele transmise / recepționate pe rețea.
- Cameră de așteptare – După ce jucătorul a selectat să joace în rețea, v-a intra apăsarea pe una din pozițiile de așteptare notate de la 1 la 4, în funcție de ordinea conectării la server, apoi fiecare jucător poate alege culoarea cu care să înceapă și trebuie să bifeze căsuța "Gata" din dreptul său pentru a porni timer-ul de 5 secunde, timp în care jucătorii se pot răzgândii și pot să debifeze căsuța pentru face modificări în legătură cu culoarea piesei.

2 Funcționalitate

2.1 Descriere

Transmiterea parametrilor către piesă.

Această funcționalitate îi permite piesei să acceseze zona de memorie în care se află informații utile pentru acțiuni precum mutare și highlight.

Mutarea pieselor pe tablă.

Pentru a muta piesele pe tablă trebuie mai întâi să fie tura ta și să fi dat cu zarul. Apoi se verifică dacă ai dat click pe o piesă care se află la final sau la bază, apoi se mută piesa pas cu pas până în poziția finală, transmițând pe rețea poziția finală și un identificator, se verifică starea caselor și dacă trebuie să scot vreo piesă a altui jucător, apoi se deblochează zarul pentru a putea da următorul jucător.

Colorarea cercului pe poziția unde v-a ajunge piesa în urma mutării.

Pentru a putea face acest lucru, se verifică tura jucătorului și poziția piesei, apoi se modifică proprietatea de culoare a cercului corespunzător poziției în care piesa v-a ajunge și salvăm într-o variabilă acel cerc pentru a-l putea reface ulterior.

Camera de așteptare.

Camera de așteptare reprezintă un loc în care jucătorii se "întâlnesc" și își aleg culoarea cu care vor să joace. După ce jucătorul a făcut alegerea, trebuie să apese pe butonul "Gata" din dreptul său pentru a le transmite celorlalți jucători că este gata pentru a începe jocul. Când toți jucătorii au confirmat că sunt gata, jocul poate să înceapă.

2.2 Fluxul de evenimente

2.2.1 Fluxul de bază

Transmiterea parametrilor către piesă

Jocul are ca parametri principali următorii: starea zarului, valoarea zarului, numărul de jucători, tura, tura de început, poziția, contor pentru final, vectori de cercuri pentru baza, final și tablă. Inițial acești parametri au valori nule, exceptând vectorii de cercuri, ale căror variabile sunt atribuite individual o singură dată la inițializarea jocului.

Transmiterea parametrilor se face imediat după crearea piesei, se transmite mai întâi identitatea piesei, fiecare piesă știe ce culoare și număr are pentru ușurința codificării/decodificării mesajelor pe rețea, apoi se atribuie valoarea turei de început a piesei.

După aceea se alocă piesei event la efectuarea unui click pentru a putea muta, apoi se atribuie valoarea 0 într-o zonă de memorie destinată contoarelor de final, se verifică dacă piesa a fost creată și se atribuie membrilor piesei adresele de memorie pentru valoare zar, starea zarului, contor final, poziția pieselor prietene, caseta de text și numărul caselor, care v-or fi prelucrate ulterior în metodele piesei. Se alocă piesei și event la mișcarea mouse-ului deasupra piesei pentru funcționalitatea de highlight.

După ce piesele au fost create și o parte din parametri transmiși către piesă, urmează să dăm click pe butonul de Zar, care ne afișează animația potrivită valorii zarului și un sunet. Apoi se transmite către piesă adresa de memorie a poziției din vector, relativă la poziția piesei la care se adaugă valoarea zarului, a cercului de tablă unde urmează să fie mutată piesa și adresa de memorie a poziției din vector, relativă la contor-ul de final al piesei, a cercului de final, și adresa de memorie a cercului unde jucătorul iese din casă " a->puneCercuri(cercuriGalben[a->iaPoz()+fJoc->valZar], cercuriFinalGalben[fJoc->finalGalben], cercA); ". Apoi se trimite piesei, dacă există, adresa de memorie a turei.

Observatorul, în urma acestei funcționalități, nu o să vadă nimic pe ecran.

Mutarea pieselor pe tablă.

Pentru ca jucătorul să poată muta piesa lui, trebuie să aștepte să-i vină tura, aceasta se stabilește pentru online în ordinea conectării la server, sau offline în ordinea apăsării butoanelor de alegere a culorii. Jucătorul trebuie să dea click pe zar înainte de a putea muta, pentru a vedea numărul de deplasări pe care le poate face.

Pieseii i-a fost atribuit după creere, un event la apăsarea ei. În momentul în care se apasă pe piesă, se apelează event-ul clasei de bază piesei numită `Pozitie`, event-ul apelează o metodă, care la rândul ei apelează o metodă virtuală numită `Joacă`, cu "pseudo-parametrii" `valZar`, starea zarului, contor final și butonul de zar care pointează către adresele de memorie în care se află defapt acei parametrii.

Metoda `Joacă` este virtuală, se face legarea dinamică a piesei cu noua clasă a jucătorului, denumită sugestiv după culoarea pieselor, și se apelează metoda din clasa derivată, clasa culorii piesei jucătorului (de ex, dacă se dă click pe o piesă galbenă, se apelează metoda `Joaca` din clasa `Galben`), aici se verifică dacă este tura ta și poți să muți, dacă nu, se afișează un mesaj sugestiv în "chat", dacă este rândul tău, dar nu ai dat încă pe zar, la click pe piesă nu se v-a întâmpla nimic, același lucru și dacă se face click pe o piesă aflată la final. Dacă piesa se află în casă când se face click pe ea, și valoarea zarului este 6, piesa v-a ieși pe prima poziție pe tablă și v-a transmite pe rețea faptul că a mutat, în caz contrar, nu se v-a întâmpla nimic.

Dacă s-a trecut de aceste verificări, se salvează poziția inițială într-o variabilă locală, și se intră într-o buclă care are rolul de a muta piesa câte o poziție până la poziția finală. În această buclă se verifică dacă am parcurs toată tabla, în acest caz se apelează o metodă din clasa de bază numită `verificaDrum`, care apelează la rândul ei o metodă virtuală numită `Aseaza`, care mută piesa la coordonatele transmise de `verificaDrum`, nu am transmis însă și parametrul auxiliar, un text, care ne permite să-i spunem piesei pe ce poziție din casă se află, așadar, poziția din casă pentru această piesă este nulă).

Ne întoarcem la metoda `VerificaDrum`, creștem contorul de final al piesei cu 1, poziția cu valoarea poziției pe care am mutat, în cazul nostru final-ul se află la poziția 41, deblocăm zarul și marcăm că putem da cu el, iar apoi pornim un sunet, asincron, specific pentru intrare la final.

(afară) Ne întoarcem la metoda `Joaca` din clasa derivată și ieșim din bucla de mutari, acum verificăm dacă am dat cu zarul valoarea 6, dacă da, afișăm pe butonul zar-ului un mesaj sugestiv, deblocăm zarul, atribuim event de click pentru piesă, (a fost retras între timp pentru a nu da click pe piese în timpul acțiunii), și atribuim unei valori speciale pentru rețea faptul că acum vreau să dau cu zarul un număr random.

Dacă nu am ajuns la finalul tablei, mutăm la locația următoare față de poziția în care ne aflăm, și în acest timp, este pornit un sunet sincronizat cu acțiunea; repetăm acești pași până când am mutat de atâtea ori de cât ne-a indicat valoarea zarului. Când am ajuns în poziția finală, se marchează faptul că putem da cu zarul, se salvează poziția finală, se transmite pe rețea faptul că acea piesă a mutat, se incrementează contorul turei jucătorului, se modifică tura în funcție de valoarea zarului, se verifică casele pentru fiecare jucător dacă există, pentru fiecare piesă a jucătorului și în fiecare poziție din bază, pentru a ști fiecare piesă în ce casă și poziție din casă se află, și ce poziții din casă sunt goale.

Se verifică apoi pozițiile inamicilor jucătorului pentru fiecare piesă care nu este în casă, se verifică apoi ce jucători inamici există, dacă au piese pe tablă și dacă se află pe aceeași poziție pe tablă, relativ la observator. Se iese din buclă și se repetă pasul (afară) aflat mai sus.

Observatorul, în urma acestei funcționalități, vede pe ecran cum piesa se deplasează cu câte o unitate, timp în care este redat un sunet la fiecare pas. (bug)

Colorarea cercului pe poziția unde v-a ajunge piesa în urma mutării.

Această funcționalitate este posibilă datorită atribuirii event-ului de mișcare a mouse-ului deasupra piesei la începutul jocului și transmiterea adresei de memorie unde se află parametrii cercului, relativ la poziția piesei și valoarea zarului, în momentul în care este apăsat butonul de zar.

Se apelează event-ul din clasa derivată specifică culorii piesei, se verifică dacă s-a dat cu zarul, în caz contrar nu are ce să coloreze și se verifică și tura jucătorului, pentru a nu colora piesele altor jucători, apoi se apelează metoda `mouseMove`, cu parametrii adresele de memorie ale proprietăților cercului deasupra căruia v-a ajunge piesa pe tablă, pe final sau pe prima poziție la ieșirea din casă.

Metoda verifică în continuare dacă există o adresă a cercului căruia poate să coloreze. Dacă există, se verifică dacă piesa se află în bază și am dat cu zarul valoarea 6 pentru a putea ieși, în acest caz se v-a colora în Aqua cercul de pe prima poziție la ieșirea din casă, altfel nu se colorează acel cerc. Se verifică dacă piesa v-a ajunge la final și colorează cercul din final, relativ la contorul de final al piesei care a fost transmis la aruncarea cu zarul. Dacă piesa v-a ajunge pe tablă,

se colorează cercul pe care aceasta v-a ajunge în poziție finală și se salvează acest cerc într-un parametru numit `lastModified`, care v-a fi ulterior folosit în timer pentru a schimba culoarea Aqua în culoarea originală.

Camera de așteptare.

Această fereastră are 2 roluri, una de a pune jucătorul să aleagă dacă vrea să joace în rețea sau în afara ei, dacă optează pentru jocul în rețea, trebuie să introducă adresa sa IP, numele de utilizator și optional portul, serverul și clientul au portul implicit 2001. În casetele de edit este afișat un mesaj sugestiv care se v-a șterge singur în momentul în care vrem să modificăm textul.

După apăsarea butonului de conectare se transmit parametrii către client socket, se blochează butoanele pentru a confirma faptul că sunt "Gata" se afișează grupul Lobby, care conține alte 4 subgrupuri, câte unul pentru fiecare jucător, care conține numele jucătorului, 4 butoane pentru selecția culorii, și butonul pentru "Gata", se activează un timer pentru a trimite nume, celălalt pentru și se redimensionează fereastra pentru a putea afișa camera de așteptare în întregime.

Timerul `tTrimiteNume`, trimite numele pe rețea pentru a putea afișa în camera de așteptare numele celorlalți jucători, înainte de asta, serverul, la conectare trimite index-ul conectării jucătorilor la server, pentru a putea afișa jucătorii în cameră în ordinea conectării.

Timerul `mVerifica`, verifică ce jucător sunt, și blochează accesul la grupul de componente a celorlalți jucători, în acest fel, un jucător nu poate modifica opțiunea altui jucător, afișează ceilalți jucători dacă s-au conectat la rețea, verifică dacă o culoare a fost aleasă și o blochează pentru ceilalți jucători și verifică dacă toți jucătorii au apăsă pe căsuța "Gata".

În momentul în care toți jucătorii s-au conectat, pot începe să își aleagă opțiunile, iar mai apoi jocul poate să înceapă.

2.2.2 Fluxuri alternative

2.2.2.1 Primul flux alternativ

Fiecare clasă derivată are câțiva parametrii specifici culorii sale, aceștia sunt contorii pentru final, cercurile de final, cercurile de casă, vectorul de poziții pentru tablă, contorul turei jucătorului, identitatea.

Mutarea pieselor pe tablă.

Codul este identic pentru fiecare clasă derivată, singura diferență fiind acești parametrii specifici culorii, doar nu am vrea ca pesa Galbenă să ajungă în baza Verde.

2.2.3 Pre-condiții

Jucătorul trebuie să dea click pe butonul de conectare, să introducă datele pentru a se conecta, să aștepte să se conecteze toți jucătorii, să aleagă o culoare și să aștepte ca toți jucătorii să aleagă o culoare, apoi să bifeze căsuța "Gata" pentru a începe jocul.

După începerea jocului, jucătorul trebuie să dea click pe zar pentru a putea muta piesa și transmite adresa cercului care urmează să fie colorat

Bug-uri cunoscute:

Jocul pornește cu socket-ul activ pentru a putea afișa toți jucătorii pentru toți clienții, în caz contrar, jucătorii care s-au conectat mai târziu, nu vor putea fi văzuți.

În camera de așteptare, după bifarea căsuței "Gata", nu se mai blochează opțiunile pentru ceilalți jucători, de aceea trebuie ca toți să fi ales deja culoarea și să nu mai modifice, pentru a exclude posibilitatea de a avea 2 jucători cu aceeași culoare.

La mutarea piesei pas cu pas, rămâne o urmă a piesei în pozițiile prin care a trecut, posibil din cauza sincronizării sunetului cu mutarea zarului.

În momentul când zarul ia o valoare care a mai fost, apare pentru o secundă animația cu ultimul cadru pentru acea valoare. În urma debugg-ului am remarcat faptul că GIF-ul devenea vizibil în momentul intrării în metoda timer-ului.

2.2.4 Post-condiții

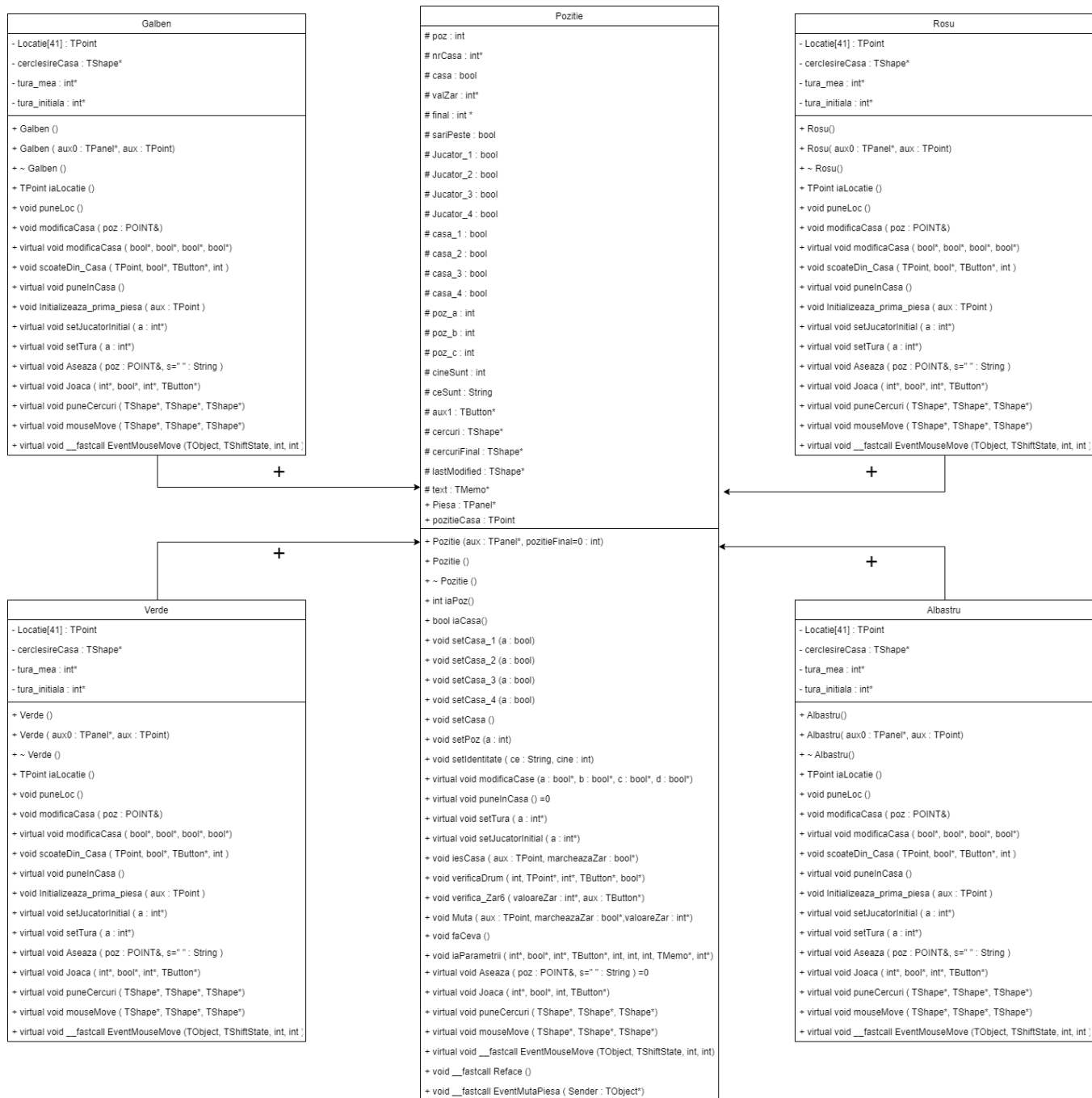
Efectuarea de click pe butoanele de alegere a culorii, sau a pornirii jocului pe rețea, care face click pe butoane în ordinea conectării la server, parametrii sunt transmiși automat către piesă. La fel și atunci când dăm cu zarul.

După ce jucătorul a dat click pe piesa pe care vrea să o mute, piesa se v-a deplasa cu câte o poziție, timp în care este redat un sunet specific mutării, deplasare care se efectuează până ce am deplasat un număr de unități egal cu valoarea zarului.

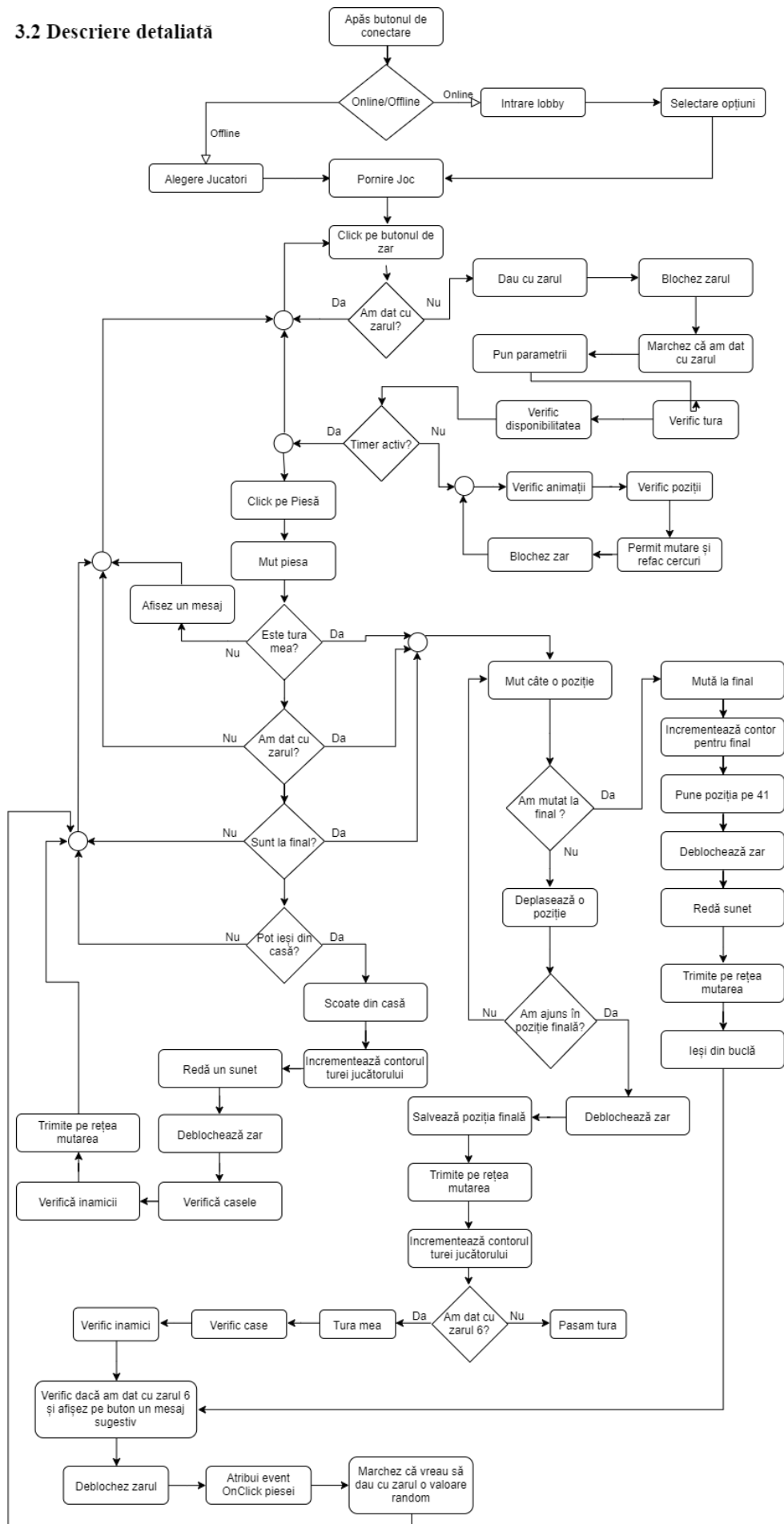
În momentul în care jucătorul mută mouse-ul deasupra piesei, cercul în care piesa v-a ajunge se v-a colora cu Aqua, iar când mută deasupra la orice alt ceva, cercul v-a revenii la culoarea originală.

3 Implementare

3.1 Diagrama de clase



3.2 Descriere detaliată



4 Bibliografie

https://www.academia.edu/22293581/PROGRAMARE_ORIENTAT%C4%82_PE_OBIECTE_%C3%8Endrum%C4%83tor_d_e_laborator

https://www.academia.edu/7121043/Borland_C_Builder_Ghid_de_initiere

<https://community.idera.com/developer-tools/general-development/>

Exemplele de jocuri create de dezvoltatori Borland din folderul de instalare al compilatorului

<https://www.youtube.com/watch?v=FZpTxuNklcA&>