Moară în rețea

23.12.2021

Îndrumător:

dr. ing. Daniel Morariu

Student:

Cotor, Otniel

(221/2)

Istoric Versiuni

Data	Versiune	Descriere	Autor
27/11/2021	0.1	Crearea tablei de joc și plasarea componentelor de tip Tbutton pe form	Cotor Otniel
1/12/2021	0.2	Crearea claselor și legăturii între acestea	Cotor Otniel
4/12/2021	0.3	Schimbarea componentelor de tip Tbutton în componente de tip Timage și afișarea pieselor ca imagini	Cotor Otniel
10/12/2021	1.0	Funcționarea mutării pieselor în funcție de tipul acestora și rundă , fără rețea	Cotor Otniel
12/12/2021	1.1	Funcționarea regulilor de luat piese în cazul unei mori, fără rețea	Cotor Otniel
17/12/2021	1.2	Implementarea rețelei	Cotor Otniel
18/12/2021	1.3	Sincronizarea între server și client a mutărilor efectuate	Cotor Otniel
19/12/2021	2.0	Implementarea regulilor de luat piese în cazul unei mori, cu rețea	Cotor Otniel

Cuprins

1 Specificarea cerințelor software

1.1 Introducere

Proiectul „Joc de moară în rețea ” este o adaptare a jocului tradițional folosind mediul de programare C++ Builder din partea firmei Embarcadero, adaptând atât regulile cât și posibilitatea de a îl juca în rețea între doi jucători.

roiectul „Joc de moara in retea” este o adaptare a jocului traditional folosind mediul de programare C++ Builder din partea firmei Embarcadero, adaptand atat regulile cat si posibilitatea de a il juca

1.1.1 Obiective

Obiectivele propuse pentru a fi realizate în acest proiect sunt mutarea unei piese , posibilitatea luării piesei adversare în cazul unei „mori”, terminarea jocului în momentul în care un jucător are doar 2 piese, implementarea polimorfismului prin modul de mutare a pieselor în funcție de ce stadiu a jocului se află , incapacitatea de mutare a piesei jucătorului în momentul în care nu este rundă lui, în cazul unei mutări interzise, capacitatea de a muta doar unde îi se permite, implementarea posibilității de a renun ța la joc dacă acesta devine repetitiv și fără sfârșit sau când câștigătorul este evident adversarul și existența unui jurnal de joc în care sunt prezente mutările jucătorilor în pozițiile respective, în mod cronologic și explicit.

1.1.2 Definiţii, Acronime şi Abrevieri

circle= poziție pe tablă în funcție de care inel al tablei se află , inelul interior având valoarea 0, cel de mijloc 1 iar exterior 2

linie= poziția pe tablă în funcție de care linie a inelului se află (asemănător unei matrici)

coloana= poziția pe tablă în funcție de care coloană a inelului se află (asemănător unei matrici)

player = jucătorul reprezentat printr-o valoare de 1 pentru jucătorul 1, respectiv de 2 pentru jucătorul 2

get+”nume_variabila”=nume de metodă care returnează valoarea dorită în funcție de parametriiset+”nume_variabila”=nume de metoda care rescrie valoarea dorita in functie de parametrii

Basic = clasa specifică tipurilor de piese care au mutări de o locație

OnlyThree= clasa specifică tipurilor de piese care rămân doar trei, acestea fără restricții de mutare

Game= clasa care gestionează schimbările și reține informațiile despre joc

set = setează valoarea unei locații în matricea 3x3x3

get= returnează valoarea unei locarii din matricea 3x3x3

tabla= matrice de 3x3x3

P1=piesele jucătorului 1

P2= piesele jucătorului 2

Index= indexul unei piese in vectorul său

round=runda

clicknr= al câtelea click

P1pieces,P2pieces= numarul de piese a fiecarui jucător

next+"nume_variabila"= incrementarea variabilei

refresh= metodă ce sincronizează valorile matricei de 3x3x3 cu valorile obiectelor piesă

transform= metodă ce transformă piesele din tip Basic în OnlyThree

i/m+circle/linie/coloana/index = variabile ce reţin valoarea variabilelor circle/linie/ coloană in anumite cazuri

1.1.3 Tehnologiile utilizate

Tehnologiile utilizate sunt mediul de programare C++ Builder Embarcadero pentru realizarea proiectului și editorul de text Microsoft Word pentru redactarea documentației . Pentru diagramele UML am folosit draw.io .

1.2 Cerințe specifice

Funcționalități ale aplicației de moară în rețea:
- terminarea jocului în momentul în care un jucător are doar 2 piese
- mutarea unei piese
- posibilitatea luării piesei adversare în cazul unei „mori"
- implementarea polimorfismului prin modul de mutare a pieselor în funcție de ce stadiu a jocului se află

2 Mutarea unei piese

2.1 Descriere

Mutarea pieselor reprezintă una din cele mai importante funcționalități , fără această jocul nu poate să continue. Mutarea pieselor se realizează prin selectarea locației de plecaer și selectoarea destinației , aceste locații fiind imagini.

2.2 Fluxul de evenimente

2.2.1 Fluxul de bază

Mutarea pieselor se realizează printr-o metodă din clasa „TGameForm" în care în funcție de număr de click- uri realizate în cadrul rundei, această va pune o piesă nouă , va lua o piesă din locul ei pentru a o mută sau chiar va pune piesă în locul dorit . Acestea se realizează prin reținerea numărului de click-uri pe tot parcursul rundei sau de rundă în care se află . În metodă , se verifică dacă sunt primele 18 runde în care piesele se plasează pe tablă , și dacă da , acestea sunt puse, și vectorii corespunzători actualizați . În cazul în care ne aflăm după rundă 18 și ne aflăm la primul click, atunci reținem poziția să în vectorul reprezentativ pentru jucătorul care mută piesă , și acesta va trebui să apese într -o altă locație pentru a comunica unde dorește să fie piesă mutată . În situația în care se realizează al 2- lea click, vom verifică dacă este corectă mutarea în nouă poziție deoarece pentru mai mult de 3 piese pentru un jucător , mutările sunt limitate la una per tură , iar dacă jucătorul deține doar 3 piese, acesta are dreptul de a mută oriunde dorește . Dacă mutarea nu este validată , piesă va rămâne unde era și rundă va trece. Dacă mutarea este validată , se vor seta proprietățile unei piese cum ar fi circle,linie și coloana pentru nouă locație , asemenea cu setarea valorii pentru poziția nouă din matricea tablă de 3x3x3. Pentru a se asigura concordanță între tablă vizuală și piesele existente , se apelează funcția refresh și afisare_tabla, care setează tablă conform tuturor pieselor existente, și afișează acea tablă în stadiul curent.

2.2.2 Fluxuri alternative

2.2.2.1 < Primul flux alternativ >

Pentru a realiza mutările pe ambele aplicații (Client și Server), după fiecare mutare(prezentată în fluxul de baza) se trimit pe rețea codficicat informațiile despre mutare respectivă , și în cazul în care se realizează moară după acea mutare, și informații despre moară . Informația (index,circle,linie, coloana) apoi este decodificată de cel în cadrul căruia nu s-a realizat mutarea și va seta proprietățile piesei cu indexul primit cu circle, linie, coloana ceea ce va realiza mutarea și pe cealaltă parte a rețelei , în final fiind din nou perechea de metode refresh și afisare_tabla.

2.2.3 Pre-condiţii

Pentru începerea aplicației , primul jucător va selecta Server, al doilea Client, iar apoi se vor realiza cele 18 plasari de piese. Odată plasate , primul va putea să mute piesele maro, specifice jucătorului 1, iar pentru jucătorul 2 în rândul sau cele albastre. Posibilitatea de a mută piesele care nu sunt ale jucătorului actual este eroare ce nu se regăsea în dezvoltarea aplicației până în cazul rețelei , însă odată prezența rețeaua , sincronizarea a fost prea dificilă , așa că fiecare jucător va trebui să încerce în a mută exclusiv piesele sale .

2.2.4 Post-condiții

După rularea funcționalității prezentate, fiecarea jucător va avea afișat pe ecran mutarea făcută de el sau de adversarul să exact în momentul realizării mutării. În cazul în care cineva dorește să mute în nouă poziției a piesei, această mutare va fi refuzată deoarece este ocupată , ceea ce verifică mutarea efectuată .

3. Implementarea polimorfismului prin modul de mutare a pieselor în funcție de stadiu a jocului se află

3.1 Descriere

În funcție de stadiu jocului, piesele se mută diferit deoarece ele sunt de tipuri diferite, însă metodă de mutare este apelată pentru fiecare piesă în parte care dorește a fi mutată , indiferent de tipul acesteia datorită prezenței polimorfisumului și implementării sale .

3.2 Fluxul de evenimente

3.2.1 Fluxul de bază

Metodă pur virtuală „Move” de tip boolean din clasa Poziție este rescrisă în clasele Basic și OnlyThree în care în funcție de aceste clase, returnează true sau false . În cazul în care obiectul este de tip Basic, mișcările sunt reduse la o poziție pe fiecare mutare, însă când vine vorba de obiecte de tip OnlyThree, acestea nu au restricții deci va returna mereu true. La începutul jocului toate piesele sunt Basic , însă când rămân doar 3, acele piese sunt înlocuite prin metodă „transform” cu obiecte de tip OnlyThree, fără a trebui să schimbăm altceva în aplicația noastră pentru a se mută special, verificările se vor face pe fiecare obiect la fel că până acum.

2.2.5 Pre-condiţii

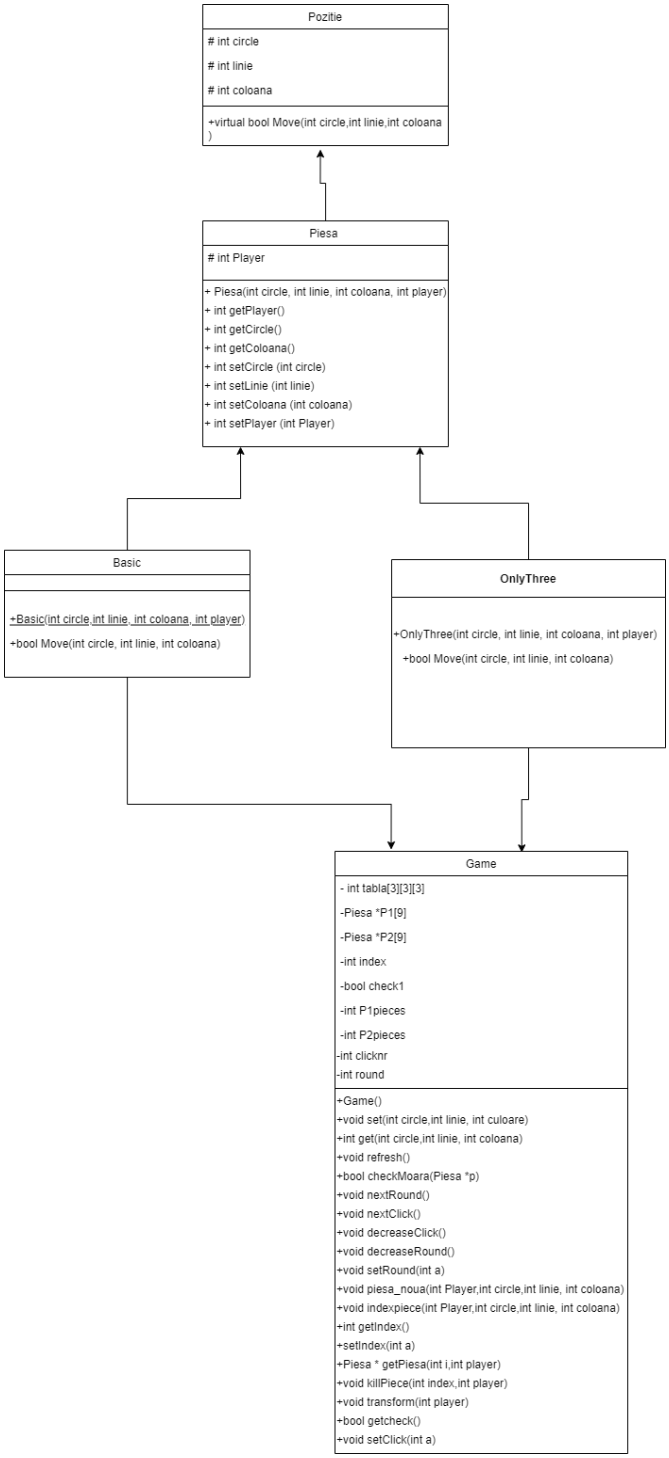
Pentru a observă funcționalitatea asta trebuie că jucătorul 1 să selecteze Server, jucătorul 2 să selecteze Client în cadrul aplicației sale , iar apoi să pună cele 9 piese de fiecare jucător , să se joace doar cu piesele sale, fără a selecta piese adverse în afară cazului unei mori când distruge piesă adversă . Până în momentul în care un jucător are mai mult de 3 piese, mutările sunt doar pe distanță de o căsuța , corespunzător regulilor jocului. În momentul în care un jucător are doar trei piese, acesta va putea să își mute piesele oriunde pe tablă .

2.2.6 Post-condiţii

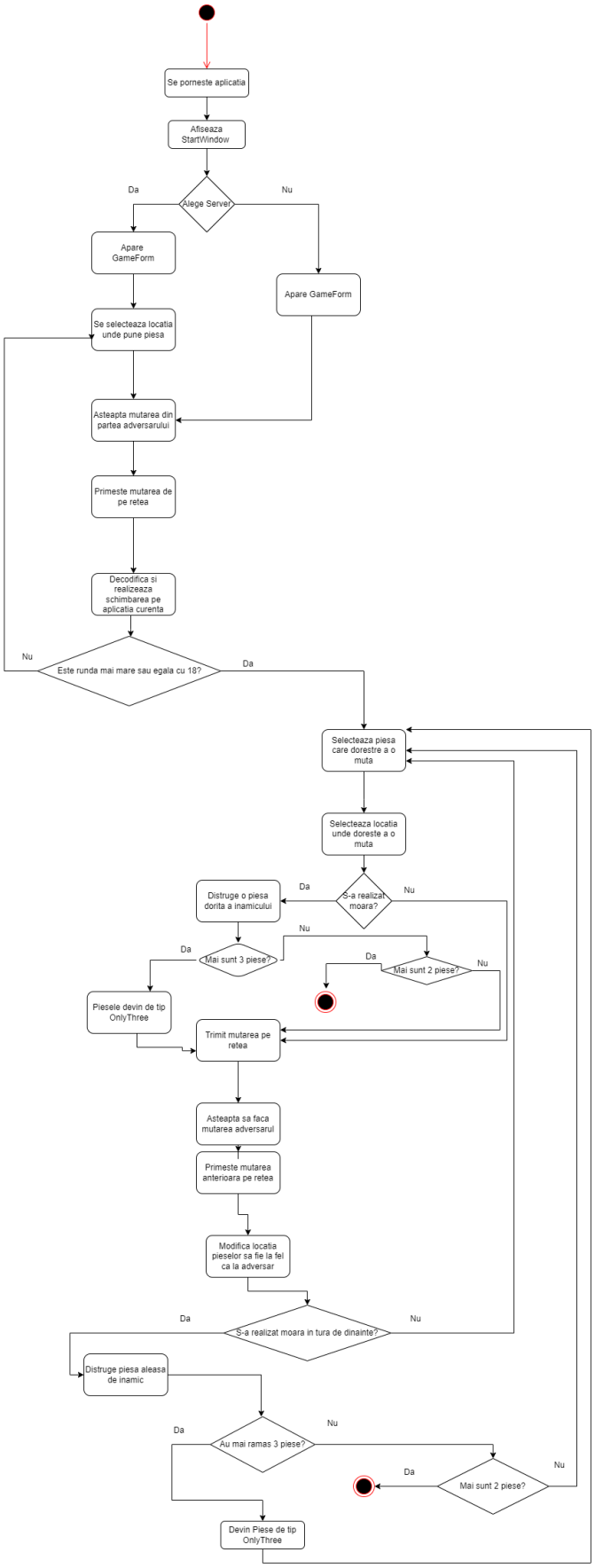
Odată realizat fluxul de la secțiunea de Pre-condiții , se poate osbserva că jucătorul cu 3 piese poate să își mute piesele oriunde pe tablă de joc, ceea ce demonstrează că pe vectorul de pointeri la Piese, în funcție de piesă , acesta se va comportă diferit însă cu același apel.

Implementare

3.3 Diagrama de clase



3.4 Descriere detaliată



4 Bibliografie

[1] Programare C++ Builder – Windows Controls http://www.functionx.com/cppbuilder/

[2]Dezvoltare aplicație cu interfață grafică - http://docwiki.embarcadero.com/RADStudio/Tokyo/en/VCL

[3] Redactarea diagramelor UML de clasa- https://www.youtube.com/watch?v=UI6lqHOVHic&ab_channel=Lucidchart

[4] Redactarea diagramelor UML de flow - https://www.youtube.com/watch?v=SWRDqTx8d4k&ab_channel=RobotRiedinger