Nu te supăra frate

08.01.2020

Îndrumător:

dr. ing. Daniel Morariu

Student:

Nistor Anamaria

224/2

Istoric Versiuni

Data	Versiune	Descriere	Autor
27.11.2019	1.0	Am creat aplicația server, clasa Piesa și clasa Pozitție, metode, contructori și am făcut moștenirea, am făcut implementarea.	Nistor Anamaria
11.12.2019	1.0	Am scris codul pentru colorarea pieselor, am făcut să apară numere random în label la fiecare apăsare a butonului btnZar.	Nistor Anamaria
13.12.2019	1.0	Am afișat cele 4 piese pe tablă.	Nistor Anamaria
14.12.2019	1.0	Am făcut piesele să se mute în căsuțele lor.	Nistor Anamaria
28.12.2019	2.0	Am mutat piesele pe poziția de start, am mai creat o metodă în clasa Piesa, am făcut ca la apăsarea butonului Play să se afișeze piesele.	Nistor Anamaria
29.12.2019	2.0	Am creat aplicația client și am scris codul de la server cu modificările specifice pentru client (al doilea jucător).	Nistor Anamaria
05.01.2020	2.0	Am schimbat pictograma aplicației.	Nistor Anamaria
06.01.2020	2.0	Am făcut ca piesa să se mute o dată și am pus o cifră pe toate piesele.	Nistor Anamaria

Cuprins

1 Specificarea cerințelor software

1.1 Introducere

Proiectul trebuie să fie jocul „Nu te supăra frate” și să fie făcut în rețea pentru 2 jucători. El trebuie să conțină pioni care vor fi mutați pe tablă conform regulilor jocului, un zar și un buton care să afișeze piesele la începerea jocului.

1.1.1 Obiective

Obiectivele pe care vreau să le realizez pentru acest proiect sunt:

- să creez un jucător;
- crearea, colorarea și afișarea pieselor de joc pentru fiecare jucător (4 cercuri);
- afișarea tablei de joc pe formă;
- crearea unui buton play care să afișeze piesele la apăsarea sa;
- crearea unei componente de tip „Label” și a unui buton care să fie zarul și care la apăsarea sa să afișeze numere random în label;
- crearea a 3 componente de tip „StaticText”, în care să scrie „Care din piese se mută”, „Cu cât se mută piesa” și „Piesle tale” și crearea a două componente de tip „Edit” în care jucătorii să scrie cifre în funcție de ce scrie în componenta de tip „StaticText” pusă deasupra sa;
- să fac piesele să se mute pe prima poziție a tablei de joc, atunci când în Label apare 6;
- să fac piesele să se mute pe tablă în funcție de cum vrea jucătorul și de regulile jocului;
- să mai creez înca un jucător;
- să fac jocul în rețea;
- să scot o componenta de tip „Edit” și să fac posibilă selectarea piesei folosind mouse-ul;
- să scot o componenta de tip „Edit” și să fac posibilă selectarea locului în care să se mute piesa, folosind mouse-ul;
- să îl fac să treacă la următorul jucător și să blocheze toate activitățile componentelor jucătorului a cărei tură a trecut;
- să fac să apară un mesaj cu „Win” la terminarea jocului;
- să schimb pictograma jocului;
- să pun imagini, în loc să desenez piesele;
- să numerotez piesele de joc.

1.1.2 Definiţii, Acronime şi Abrevieri

Poziție – numele clasei de bază, folosită pentru a seta și schimba coordonatele pieselor (cercurilor);

Set_Poz – metodă a clasei Poziție, folosită la schimbarea coordonatelor pieselor;

Set_Start – metodă a clasei Poziție, folosită la schimbarea coordonatelor pieselor;

Piesa – numele clasei derivate,moștenește clasa de bază, folosită pentru a crea piesele;

Mută – metodă a clasei Piesa, folosită la schimbarea coordonatelor pieselor (cercurilor), mută piesele în funcție de ce este afișat în label și de cum dorește jucătorul;

Start – metodă a clasei Piesa, folosită la schimbarea coordonatelor pieselor, pune piesele pe prima poziție a tablei de joc după ce label afișează 6;

Afișează – afișează, creează și colorează piesa pe tablă;

Șterge – șterge piesa afișată creând una nouă, de altă culoare

pr – piesa roșie, vector care indică fiecare piesă creată;

i,t,s și alte variabile declarate – au fost declarate în general aleatoriu sau au o mică legătură cu partea în care au fost folosite;

lText, eText și restul denumirilor componetelor – au fot numite astfel pentru a fi deosebite una de alta și a-ți putea da seama cât mai ușor la care din ele facem referire.

MAX_PIESE – modifică numărul de piese într-un sigur loc dacă este nevoie, și nu în fiecare loc în care este folosită;

1.1.3 Tehnologiile utilizate

Aplicațiile software folosite sunt: C++Builder 6 și Paint. Documentația este luată de pe:

- http://www.science.upm.ro/~traian/web_curs/Cpp/s_logice/s_logice.html;
- https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/.

1.2 Cerințe specifice

Obiectivele care au fost realizate sunt:

- să creez un jucător;
- crearea, colorarea și afișarea pieselor de joc pentru fiecare jucător (4 cercuri);
- afișarea tablei de joc pe formă;
- crearea unui buton play care să afișeze piesele la apăsarea sa;
- crearea unei componente de tip „Label” și a unui buton care să fie zarul și care la apăsarea sa să afișeze numere random în label;
- crearea a 3 componente de tip „StaticText”, în care să scrie „Care din piese se mută”, „Cu cât se mută piesa” și „Piesle tale” și crearea a două componente de tip „Edit” în care jucătorii să scrie cifre în funcție de ce scrie în componenta de tip „StaticText” pusă deasupra sa;
- să fac piesele să se mute pe prima poziție a tablei de joc, atunci când în Label apare 6;
- să mai creez înca un jucător;
- să schimb pictograma jocului;
- să schimb pictograma jocului;
- să fac piesele să se mute pe tablă (se mută doar prima dată, după care dă eroare);
- să numerotez piesele de joc (se pune aceeași cifră pe toate).

2 Funcționalitate

2.1 Descriere

Funcționalitatea pe care o consider relevantă aplicației pe care am ales să o fac este afișarea pieselor pe tabla de joc, deoarece dacă nu am piese de joc, atunci jocul nu poate începe, piesele nu pot fi mutate și jucătorii nu au ce face cu el, practic jocul nu există.

2.2 Fluxul de evenimente

2.2.1 Fluxul de bază

Această funcționalitate folosește o metodă din clasa derivată numită Piesa. Numele metodei este Afișează și e folosită la crearea unui cerc cu funcția Ellipse și un pointer *r de tip Trect, și la colorarea piesei folosind alte funcții specifice programului utilizat. Metoda este apoi apelată în implementarea altor două metode, Start și Mută, ca mai apoi să fie utilizată în crearea cercurilor pe tablă.

Utilizatorul o să vadă piesele pe tablă în momentul în care apasă butonul play. Acestea vor fi așezate direct pe pozițiile de început, mai exact pe pătrățelele care indică casa jucătorului, locul din care vor porni la începerea jocului.

Funcționalitatea o să fie apelată automat de fiecare dată când metode Start și Mută vor fi apelate în program, mai exact de fiecare dată când piesele o să se mute pe tablă.

Astfel, utilizatorul o să vadă piesele pe tablă în momentul în care apasă butonul play și de fiecare dată când mută o piesă, folosindu-se de componentele de tip Edit și de zarul din dreapta tablei. Acestea vor fi așezate direct pe pozițiile de început, mai exact pe pătrățelele care indică casa jucătorului, locul din care vor porni la începerea jocului și din care se vor muta pe rând atunci când în Label este afișată cifra 6, urmând să continue să se mute pe tablă în funcție de pătrățele.

2.2.2 Pre-condiţii

Acțiunile necesare pornirii corecte a funcționalității sunt:

- pornirea jocului;
- apăsarea butonului play o singură dată, deoarece de fiecare dată când este apăsat se va afișa un nou set de piese și în același timp vor rămâne cele care au fost mutate, ca mai apoi să nu mai funcționezecum trebuie;
- în caz de suprapunere a pieselor, va rămâne una singură.
- la mutarea pieselor, cifra ramâne în loc săse șteargă.

2.2.3 Post-condiţii

După rulare, pentru ca utilizatrul să vadă piesele este nevoie să apese play.Dacă după apăsarea acestui buton apar piesele de joc în locul în care trebuie, mai exact pe pătrățelele de aceeași culoare cu piesele. Dacă se întâmplă toate aceste lucruri, atunci înseamnă că funcționalitatea merge cum trebuie. Dacă după ce începe să mute piesele, acestea nu apar în două locuri, în cel în care au fost și în cel în care s-au mutata, atunci însemană că funcționează cum trebuie.

3 Funcționalitate

3.1 Descriere

Funcționalitatea pe care o consider relevantă aplicației pe care am ales să o fac este mutarea pieselor pe tabla de joc, deoarece dacă nu se pot muta piesele pe tablă, atunci vom avea doar piesele afișate pe tablă, iar utilizatorii nu au ce face cu programul, nu pot juca acest joc.

3.2 Fluxul de evenimente

3.2.1 Fluxul de bază

Această funcționalitate folosește două metode din clasa derivată numită Piesa și alte două moștenite din clasa de bază numită Poziție. Numele metodelor sunt Mută, Start, Set_Poz și Set_Start, și sunt folosie la modificarea coordonatelor pieselor (cercurilor) afișate pe tabla de joc. Set_Poz și Set_Start sunt apelate în implementările metodelor Mută și Start, pentru a fi utilizate atunci când acestea sunt apelate în program.

Utilizatorul o să vadă piesele pe tablă în momentul în care apasă butonul play. Acestea vor fi așezate direct pe pozițiile de început, mai exact pe pătrățelele care indică casa jucătorului, locul din care vor porni la începerea jocului.

Pentru a fi așezate astfel, este apelată metoda Start. Tot metoda Start este apelată și atunci când zarul indică 6 și utilizatorul scoate o piesă din casa sa. Pentru a muta piesele pe tabla de joc, este folosită metoda Mută.

Astfel, utilizatorul vede piesele mutându-se pe tablă conform a ceea ce indică în componentele de tip Edit și a zarului.

3.2.2 Fluxuri alternative

3.2.2.1 < Primul flux alternativ >

Inițial am început doar cu metodele Mută și Set_Poz, dar pe parcurs am decis să mai adaug încădouă metode, Start și Set_Start, deoarece am considerat că este mult mai ușor și mai sigur să fac așa. Diferența celor două este în implementarea metodelor Set_Poz și Set_Start. În prima metodă menționată x și y sunt adunate separat și de fiecare dată cu o valoare indicată de mine, cu dx sau dy, în timp ce a doua metodă ia de fiecare dată valoarea indicată de mine, x ia valoarea lui dx și y pe cea a lui dy, astfel fiind mult mai utilă în cazul în care este necesar ca piesele să aibă coordonate fixe.

3.2.3 Pre-condiţii

Acțiunile necesare pornirii corecte a funcționalității sunt:

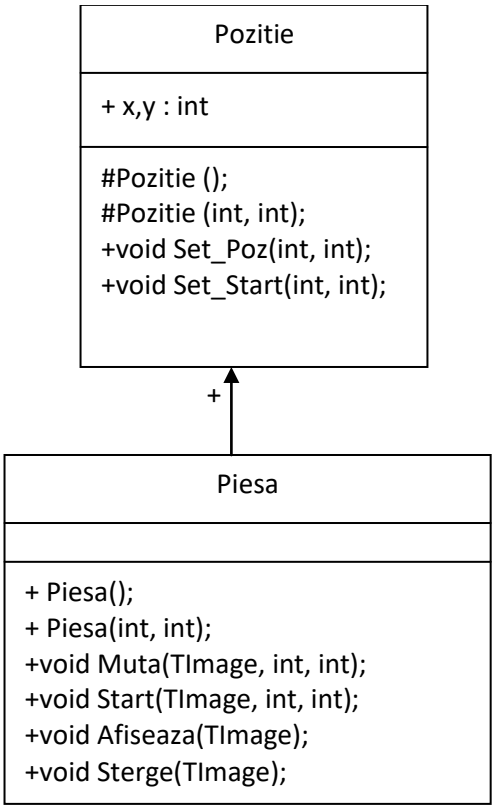
- pornirea jocului;
- apăsarea butonului play o singură dată, deoarece de fiecare dată când este apăsat se va afișa un nou set de piese și în același timp vor rămâne cele care au fost mutate, ca mai apoi să nu mai funcționezecum trebuie;
- să apese butonul „Aruncă zarul” până când acesta afișează 6 deasupra sa, iar apoi să scrie în locul indicat numărul piesei pe care dorește să o scoată din casă;
- o să apese pe zar încă odată, iar apoi o să scrie în locul indicat cifra care apare deasupra zarului pentru a muta piesa;
- nu o să facă ce trebuie, deorece nu am putut pune condițile astfel încât să se mute cum trebuie piesa;
- la următoarea aruncare de zar programul nu o să mai facă nimic, iar dacă încercăm să scooatem altă piesă, acesta vada eroare.

3.2.4 Post-condiţii

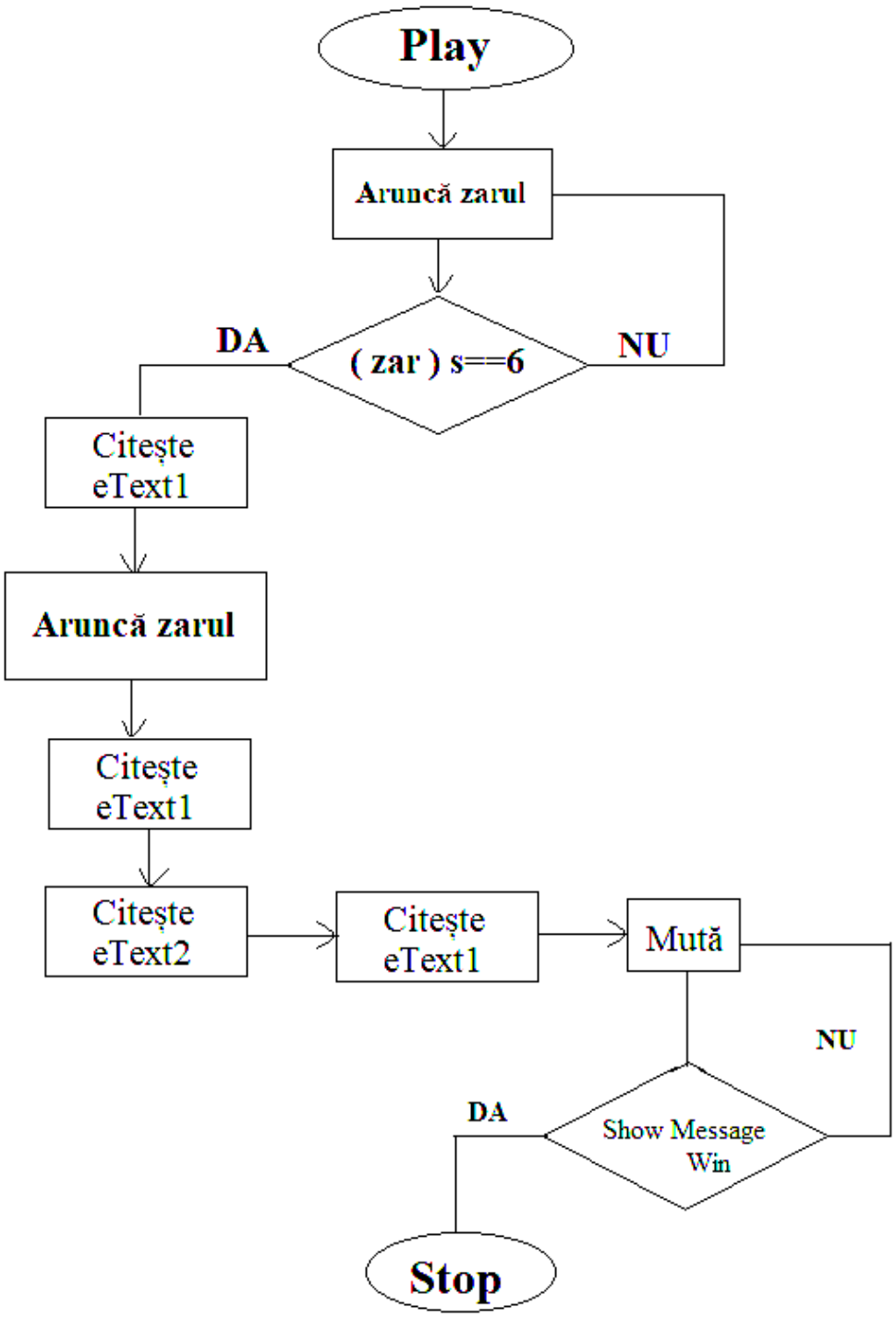
După rulare utilizatorul vede piesele puse pe pătrățelele de aceeași culoare. În cazul în care apasă pe butonul pentru aruncarea zarului și apare deasupra sa cifra 6, iar apoi scrie în componenta de tip Edit pe care piesă vrea să o scoată din casă, o să observe că piesa s-a mutat pe poziția de start care aparține acelei culori. După aceea poate observa cum după următoarea aruncare a zarului și indicarea piesei și a numărului de căsuțe, piesa se mută pe tablă. Dacă programul face totul corect până la scoaterea unei piese, atunci înseamnă că o parte din funcționalitate merge cum trebuie. Dacă și în cotinuare face ce trebuie, atunciînsemnă că toată funcționalitatea este corectă, și nu doar o parte din ea.

4 Implementare

4.1 Diagrama de clase



4.2 Descriere detaliată



5 Bibliografie

1. yevol.com/bcb/Lesson12.htm;
2. https://classroom.google.com/u/1/c/MzgzNTI4MDMwNDRa/m/NDMyNzE4NjEyOTBa/details;
3. https://www.google.ro/search?q=tabla+nu+te+supara+frate&sxsrf=ACYBGNSyiGKdxXKA9d8TE6bcQjpkpm-q0g:1577970788463&source=lnms&tbm=isch&sa=X&ved=2ahUKEwissr-i_TmAhVhtIsKHXUmBuYQ_AUoAXoECAsQAw& biw=1366&bih=657#imgrc=KibVuc2VsHaDfM: .