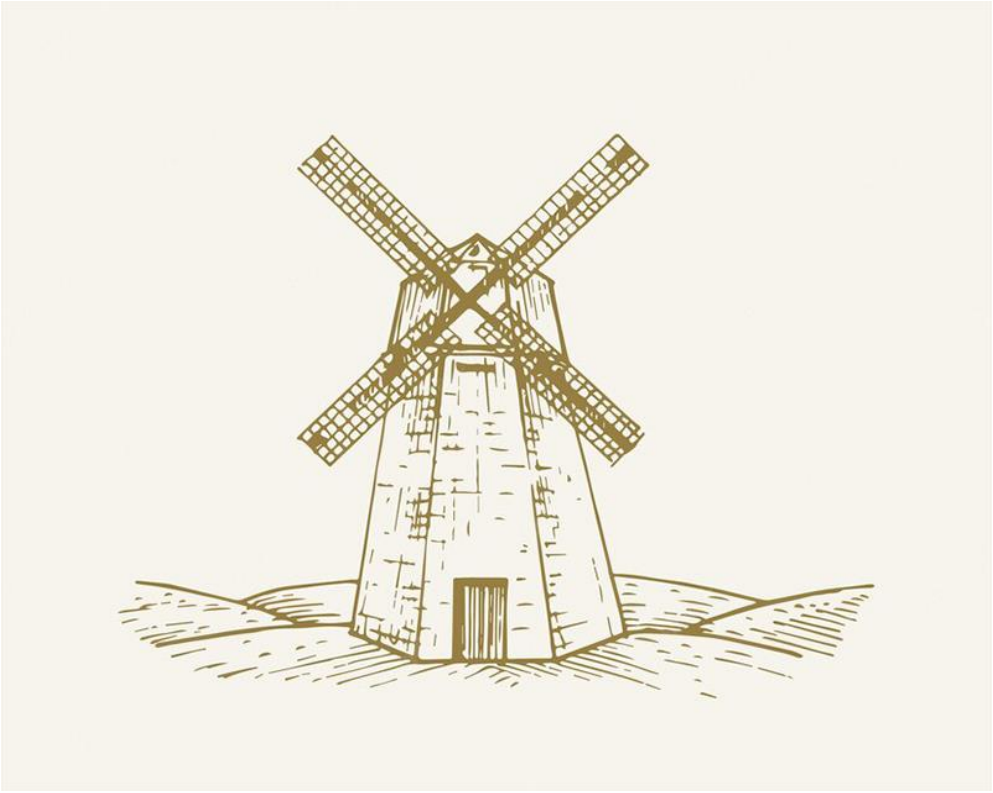
Joc Moară

14.01.2023



Îndrumător:

dr. ing. Daniel Morariu

Student: Stoica Catalin-Constantin

Tehnologia Informatiei 223/2

Istoric Versiuni

Data	Versiune	Descriere	Autor
28/10/2022	1.0	Creearea unui meniu principal care permite alegerea opțiunii de a începe jocul sau de a vedea regulile jocului de moară	Stoica Catalin
29/10/2022	1.1	Creearea ferestrei „Player Register” care permite jucătorilor să își introducă numele	Stoica Catalin
29/10/2022	1.2	Creearea ferestrei în care se desfășoară efectiv jocul	Stoica Catalin
11/11/2022	2.0	Definirea clasei Board(colecție de date despre poziția unei piese pe tablă) Definirea clase Game (clasa care se ocupă de inițializarea obiectelor de tip Board cât și de setările începutului de joc	Stoica Catalin
12/11/2022	2.1	Implementarea butoanelor rotunde care reprezintă piesele de joc	Stoica Catalin
18/11/2022	3.0	Creeare clasei Mill (permite așezarea pieselor, ștergerea acestora și verifică când se realizează moara)	Stoica Catalin
19/11/2022	3.1	Adăugarea în cadrul clasei Mill a metodelor care permit jucătorului să așeze piesa, să verifice dacă aceasta formează moara iar daca acest lucru se intamplă, să șteargă o piesă oponentului (place, millVerifyRow,deletePiece)	Stoica Catalin
25/11/2022	3.2	Adăugarea în cadrul clasei Mill a metodelor care permit jucătorului să verifice dacă poate muta piesa, mutarea piesei cât și ștergerea unei piese adverse dacă acesta realizează moara(movePiece, verifyPos, deleteCurrentPosition)	Stoica Catalin
3/01/2023	4.0	Implementarea principiilor de moștenire și derivare a POO prin creearea clasei MillMoves și a metodei deletePiece	Stoica Catalin
3/01/2023	4.1	Adăugarea funcționalităților claselor în cadrul acțiunii asupra butoanelor	Stoica Catalin
6/01/2023	4.2	Desemnarea câștigatorului prin implementarea metodelor de verificare	Stoica Catalin

Cuprins

ISTORIC VERSIUNI **2**

CUPRINS ..... ERROR! BOOKMARK NOT DEFINED.

1 SPECIFICAREA CERINȚELOR SOFTWARE **4**
1.1 Introducere **4**
1.1.1 Obiective 4
1.1.2 Definiții, Acronime și Abrevieri 5
1.1.3 Tehnologiile utilizate 5
1.2 Cerințe specifice **6**

2 AȘEZAREA PIESELOR PE TABLĂ **7**
2.1 Descriere **7**
2.2 Fluxul de evenimente **7**
2.2.1 Fluxul de bază 7
2.2 Pre-condiții 9
2.3 Post-condiții 9

3 ȘTERGEREA PIESELOR DE PE TABLĂ **9**
3.1 Descriere **9**
3.2 Fluxul de evenimente **9**
3.2.1 Fluxul de bază 9
3.2 Pre-condiții 10
3.3 Post-condiții 10

4 IMPLEMENTARE **11**
4.1 Diagrama de clase **11**
4.2 Descriere detaliată **12**

5 BIBLIOGRAFIE **13**

1 Specificarea cerințelor software

1.1 Introducere

Proiectul meu constă în aplicarea cunoștințelor de programare orientată pe obiect pentru a realiza clasicul și binecunoscutul joc de Moară. Moară (cunoscut și sub numele de țintar sau car) este un joc de societate jucat în doi, din categoria jocurilor de strategie. Tabla de joc are desenate trei pătrate concentrice și conectate la mijlocul laturilor, ca în figura alăturată. Colțurile dreptunghiurilor și mijloacele laturilor sunt marcate cu puncte care arată cele 24 de poziții în care se pot plasa cele 18 piese, câte nouă de fiecare jucător.

1.1.1 Obiective

Obiectivul proiectului a fost de a creea o aplicație perfectă pentru relaxare dar care stimulează și gândirea în același timp. Am ales acest tip de proiect din pasiunea pentru jocuri de strategie cât și nostalgia pe care acest joc clasic o aduce.

Obiectiv propus	Obiectiv realizat
Crearea interfețelor: • O interfață care să prezinte meniul jocului • O interfață în care jucătorii își pot introduce numele și genul • O interfață în care se desfășoară jocul efectiv • Mesajul care te avertizează că urmează să părăsești jocul	
Posibilitarea jucătorilor de a-și introduce numele cât și selectarea genului	
Structurarea proiectului conform cerințelor (clase, metode, principii de programare)	
Așezarea pieselor pe tablă conform acțiunii cu mouse-ul asupra butoanelor	
Colorarea butoanelor corect conform rândului în care jucătorii pot acționa	
Posibilitatea de a șterge o piesă adversă atunci când se formează moara (la prima așezare)	
Posibilitatea jucătorilor de a-și muta piesele	
Afișarea pozițiilor posibile în care jucătorul poate să își mute piesa	
Posibilitatea de a șterge o piesă adversă atunci când se formează moara	

Afișarea unui textbox care să indice ordinea în care rândul jucătorilor se schimbă cât și acțiunile pe care aceștia trebuie să le faca	
Afișarea numelui câștigătorului la final de joc	
Crearea rețelei (pentru multiplayer)	
Jucătorii să nu poată scoate o piesă din cadrul unui grup care formează moara pentru adversar (conform regulamentului)	
Adăugarea unui timer pentru mișcări	
Jucătorii să aibă opțiunea de a muta piesele în orice poziție liberă atunci când aceștia rămân cu 3 piese pe tablă („ moara săltăreață” confrom regulamentului)	

1.1.2 Definiţii, Acronime şi Abrevieri

Pe lângă crearea funcționalităților jocului și utilizarea principiilor de POO, am încercat să folosesc principiile de Clean Code (DRY, YAGNI, KISS) utilizând denumiri sugestive/simple și metode clare, scurte care nu fac mai multe lucruri. Pentru denumirea metodelor si a variabilelor am folosit practica „Camel Case”.

Scop: scrierea unui cod ușor de citit.

Reguli generale de denumire a termenilor folosiți:

- Metode: verbe care încep cu literă mare;
- Variabile: substrantive care încep cu literă mica;
- Clase: substantive care încep cu literă mare.

Variabile care nu au un nume sugestiv: del , ok, m, x,

1.1.3 Tehnologiile utilizate

In realizarea proiectului s-au utilizat:

- Visual Studio 2019 – mediu de programare
- Adobe Photoshop CC 2019 – pentru editarea fundalului meniului principal si a avatarelor jucatorilor
- Microsoft Word – pentru scrierea documentației
- Clasa Roundbutton (site-ul stackoverflow) -pentru creeare butoanelor custom

1.2 Cerințe specifice

- Prima mișcare a jocului revine jucătorului în mod aleatoriu
- Așezarea pieselor pe tablă
- Verificarea dacă se realizează moara
- Posibilitatea de a alege o piesă adversă care să fie scoasă din joc
- Mutarea pe rând a pieselor pe tablă
- Mesajele și indicațiile pe care jocul le transmite jucătorilor
- Desemnare câștigător

2. Așezarea pieselor pe tablă

2.1 Descriere

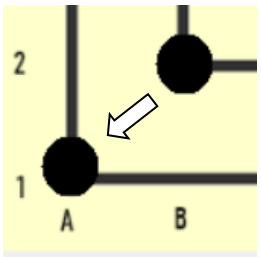
Funcționalitatea pe care se bazează proiectul este acțiunea mouse-ului prin click asupra butoanelor. Acestea reprezintă piesa propriu-zis iar o acțiune de click asupra unui buton poate însemna selectarea piesei pe care o mutăm, mutarea într-o poziție nouă sau ștergerea unei piese adverse .

2.2 Fluxul de evenimente

2.2.1 Fluxul de bază

Clasa principală prin intermediul căreia se realizează acțiuni specifice jocului este clasa **Mill**. Metodele definite aici și apelate prin evenimentul de tip click, realizează mișcările și consecințele acestora pe tablă.

Butoanele sunt astfel denumite pentru a putea intui poziția lor pe tablă. Spre exemplu avem :



1A reprezintă poziția pe tablă dar și poziționarea în matricea cu coordonate de tip ***Board***

```
private void btn1A_Click(object sender, EventArgs e)
        {
            if (m.delPiece == true) //semafor care permite executarea stergerii
            {
                m.deletePiece(6, 0, btn1A); //stergerea piesei de pe tabla
                render(); //actualizeaza schimbarile de pe table (stergerii, indicatii noi)
            }
            else
            {
                m.place(6, 0, btn1A); //asezarea piesei pe tabla
                m.verifyPos(3, 0, btn4A); // verifica daca pozitia este disponibila mutarii
                m.verifyPos(6, 3, btn1D); // se observa ca parametrii sunt coordonatele poz. vecine
                m.deleteCurrentPosition(6, 0, btn1A); // sterge piesa din pozitia curenta
                m.movePiece(6, 0, btn1A); // se executa atunci cand celelalte functionalitati nu au loc si realizeaza mutarea piesei conform coord.
                render();
            }


        }
```

În continuare avem metoda **place** din cadrul clasei **Mill**. Aceasta realizează așezarea pe tablă a unei piese (acest proces reprezintă o primă parte a jocului).

```
public void place(int x, int y, WindowsFormsApplication1.RoundButton roundButton)
    {

        if (board[x, y].pos != 1 && (p1Pieces != 0 || p2Pieces != 0))
        //se verifica daca pozitia este libera si daca jucatorii mai au piese de asezat pe tabla
        {
            if (p1 == 1) //daca este randul primului jucator
            {

                pieceRedOrBlue(x, y, roundButton); // metoda care in functie de randul jucatorului decide daca pe table se pune o piesa rosie sau una albastra
                p1Pieces--; //nr de piese de asezat pe tabla este decrementat
                indicationText = "Randul jucatorului albastru !"; //textbox-ul cu informatii este modificat
                if (millVerifyRow(x, y) == 1 || millVerifyCol(x, y) == 1) // se verifica daca se formeaza moara, daca este afirmativ se schimba din nou texbox-ul, nr efectiv de piese ramase se modifica cat si un semafor permite mai apoi executarea metodei de stergere
                {
                    indicationText = "MOARA ROSIE ! Alege o piesa albastra pe care sa o scoti!";
                    inGameP2Pieces--;
                    del = true;
                }


            }
            else // aceleasi proceduri ca si mai sus pentru jucatorul albastru (player 2)
            {

                pieceRedOrBlue(x, y, roundButton);
                p2Pieces--;
                indicationText = "Randul jucatorului rosu !";
                if (millVerifyRow(x, y) == 1 || millVerifyCol(x, y) == 1)
                {
                    indicationText = "MOARA ALBASTRA ! Alege o piesa rosie pe care sa o scoti !";
                    inGameP1Pieces--;
                    del = true;


                }
            }
        }
```

2.2 Pre-condiţii

Utilizatorul aplicației va avea de ales între a începe procesul de începere al jocului prin apăsarea butonului de Start sau de a vedea regulile jocului de moară prin apasarea butonului Reguli . În cazul în care este acționat butonul Start se va deschide o fereastră în care cei 2 jucători se pot înregistra prin introducerea numelui și selectarea genului (M/F – avatar care va fi alăturat numelor). Și această fereastră are la rândul ei un buton de start care deschide fereastra în care se va desfășura jocul propriu-zis.

În dreapta tablei de joc se află un textbox care afișează de fiecare dată ceea ce trebuie să facă jucatorii . Am adăugat acest textbox întocmai ca regulile jocului să fie respectate, astfel încât utilizatorii să știe rândul cui este și acțiunea să aibă un rezultat previzibil

Prima mutare a jocului revine jucătorului într-un mod random la fiecare start nou de joc.Și acest lucru este menționat în textbox-ul cu indicații .

2.3 Post-condiţii

Jucătorul observă că fundalul butonului apăsat se colorează din negru în culoarea de care este reprezentat jucătorul (roșu pentru player1 albastru pentru player2). În cazul în care prin plasarea piesei se formează o moară, jucătorul este atenționat de textbox-ul cu indicații că trebuie să scoată o piesă a adversarului ca jocul să continue.

3. Ștergerea pieselor de pe tablă

3.1 Descriere

O altă funcționalitate esențială în execuția corectă a programului și desfășurarea jocului este cea de ștergere a unei piese. Metodele de ștergere sunt apelate în cadrul unui eveniment de tip click asupra unui buton. Acestea se apelează în cazul în care „semafoarele” sunt pornite, mai concret dacă se formează moara .

3.2 Fluxul de evenimente

3.2.1 Fluxul de bază

Ca și exemplu , funcția inițială de ștergere . Aceasta este declarată ca fiind virtual pentru a putea folosi mai apoi principiile de derivare și polimorfism ale POO.

```
public virtual void deletePiece(int x, int y, WindowsFormsApplication1.RoundButton roundButton)
    {
        roundButton.BackColor = System.Drawing.Color.Black; //culoarea butonului redevine neagra
        board[x, y].pos = 0; // se semnaleaza faptul ca pozitia este goala
        board[x, y].xpos = 0; // coordonata x devine 0
        board[x, y].ypos = 0; //cordonata y devine 0
        board[x, y].pcolor = 'q'; // caracterul care indica culoarea pozitiei din matrice devine „inutila”
        del = false; // un semafor care nu permite de 2 ori executarea metodei de stergere

    }


public override void deletePiece(int x, int y, WindowsFormsApplication1.RoundButton roundButton)
    {
        if (delPiece == true) // daca se intra pe aceasta ramura, textboxul se va modifica  in functie de desfasurarea normala a jocului
        {
            if (roundButton.BackColor == System.Drawing.Color.Red)
                base.indicationText = "Jucatorul rosu muta o piesa !"; //textboxul se modifica
            else
                base.indicationText = "Jucatorul albastru muta o piesa !"; //textboxul se modifica

            base.deletePiece(x, y, roundButton); //se apeleaza metoda de stergere din clasa de baza
            delPiece = false; // un semafor care nu permite de 2 ori executarea metodei de stergere
        }
        else
            base.deletePiece(x, y, roundButton);  //se apeleaza metoda de stergere din clasa de baza fara a modifica continutul textboxului

    }
```

Metoda **deletePiece** din clasa moștenită se apeleaza în momentele în care se realizează moara în urma mutării pieselor . Metoda din clasa de bază se apelează în cadrul altor metode precum cea prin care poziția curentă se șterge înainte de a muta piesa sau în care se formează moara prin așezarea directă a pieselor .

3.3 Pre-condiţii

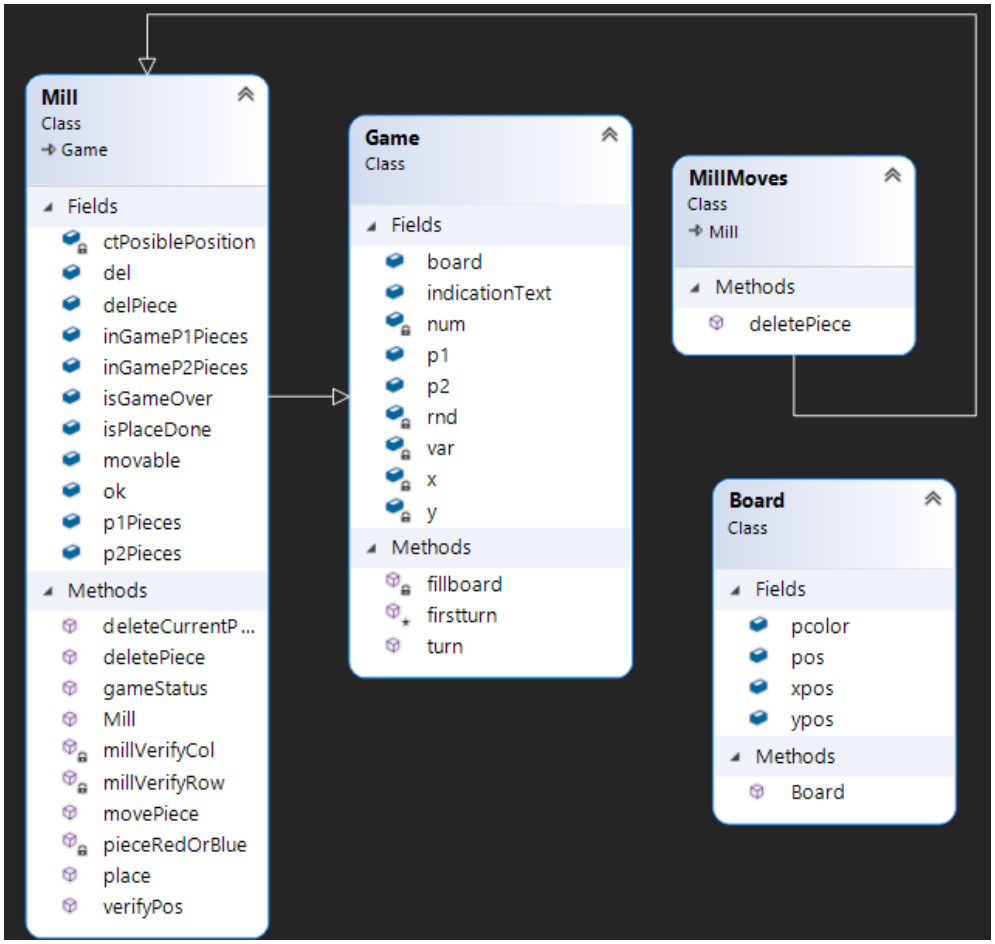
Jucătorul care realizează moara este atenționat de către textbox că trebuie să scoată din joc o piesă adversă . Acesta alege după bunul plac piesa pe care o va scoate și o va selecta printr-un click pe butonul respectiv . În acest fel aceea poziție va deveni liberă iar jocul își va continua cursul.

3.4 Post-condiţii

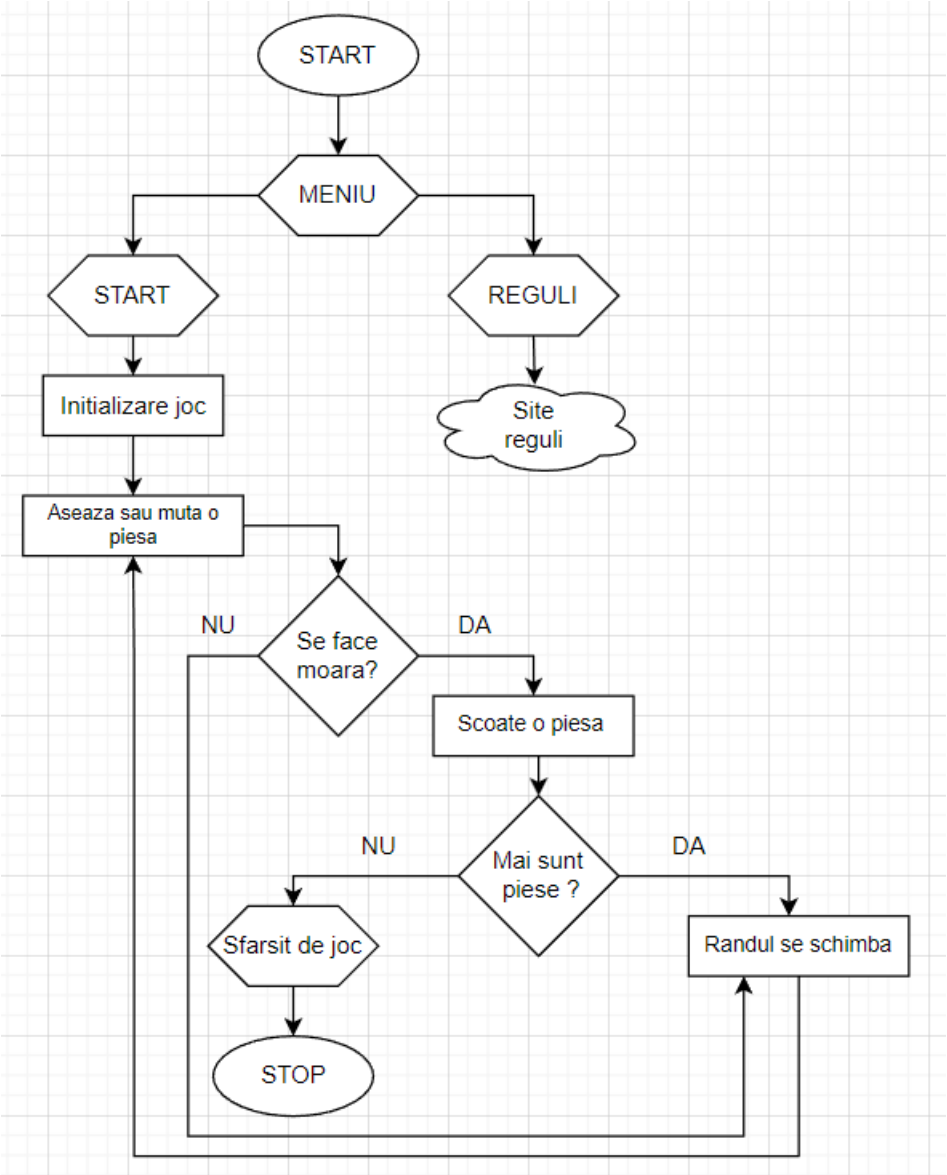
De pe tablă va dispărea o piesă. Poziția acesteia va deveni „goală” și disponibilă pentru eventuale mișcări . Numărul de piese al adversarului se va decrementa, jocul continuând în mod normal în cazul în care numărul de piese este mai mare dect 2.

4 Implementare

4.1 Diagrama de clase



4.2 Descriere detaliată



5 Bibliografie

https://visualstudio.microsoft.com/downloads/
https://www.cumsejoaca.ro/reguli-si-regulamente/regulile-jocului-de-moara-tintar/
https://stackoverflow.com/questions/3708113/round-shaped-buttons
https://app.diagrams.net/?src=about
https://stackoverflow.com/questions/13536993/constructor-parameters-and-inheritance
https://www.educba.com/c-sharp-object-sender/
https://www.codeproject.com/Questions/312402/How-to-Connect-2-forms-in-Csharp-windows-applicati