**Proiect Jocul Lui Nim**

**Descriere:**

Jocul lui Nim este un joc mathematic de strategie,care se joaca in 2 jucatorii astfel:

Se considera 5 gramezi de pietre prima gramada cu o piatra,a doua gramada cu

2 pietre a 3 cu 3 pietre s.a.m.d.

Un Jucator poate selecta oricate pietre atata timp cat ele sunt pe aceeasi coloana.

Nu are voie sa selecteze pietre care nu sunt in aceeasi gramada.Castiga jucatorul care trage ultima piatra.

**Clasele:**

In cadrul acestui proiect am folosit urmatoarele clase:

* Graphics
* Player
* DissapearActionListener
* NewGameActionListener

**Clasa Graphics:**

Clasa Graphics tine loc de View in cadrul proiectului si in ea declaram componentele de interfata grafica JFrame si JPanel, pe langa astea mai folosim in proiect si componenta JButton.In cadrul clasei setam dimensiuniile si coordonatele fiecarei componenta.

**Clasa Player:**

In clasa Player,care tine loc de Model care are 2 atribute Nume si Turn.

Turn reprezinta daca este sau nu randul lui la mutare.

**Clasa DissapearActionListener:**

In clasa DissapearActionListener,care tine loc de Controller avem actiuniile care pot avea loc asupra butoanelor din joc.

**Clasa NewGameActionListener:**

In clasa NewGameActionListener lucram pentru butonul New Game care odata ce este apasat seteaza toate butoanele initiale pe vizibil.

**Implementare proiect:**

Se porneste de la o fereastra pe care sunt asezate strategic 15 butoane setate pe isVisible(true).Fiecare jucator poate selecta un annumit numar de butoane in functie de coloana pe care o alege.Selectarea butoanelor care pot fi alese deodata se face in functia mouseEntered acestea fiind retinute intr-un arraylist de JButton-uri.La apasarea unuia dintre butoane toate butoanele din arraylist sunt puse pe isVisible(false).Daca mouse-ul iese de pe buton fara sa faca click butoanele din arraylist se sterg din cauza functiei mouseExit.Jucatorii se schimba alternativ la fiecare eveniment click.De fiecare data verificam daca jocul s-a terminat, in acest caz afisam o fereastra cu dialog care continue numele jucatorului castigator.Stim ca jocul a fost terminat in momentul in care toate butoanele au fost apasate.Dupa afisarea mesajului se reincepe jocul automat prin apasarea automata a butonului click.

**Metode:**

**1)Metoda changeTurn()**

->este folosita pentru a schimba randul jucatorului

2)**Metoda actionPerformed**

**->**mostenita din interfata ActionListener si este folosita pt butonul NewGame: cand este apasat se reseteaza jocul

3)**Metoda endGame**

->este o functie care returneaza daca jocul s-a terminat sau nu in functie

Daca butoanele apasate au devenit toate pe isVisibile(false)

3)**Metoda mouseEntered**

**->**este o metoda mostenita care ne avertizeaza cand mouseul se afla deasupra butonului sursa.Cu ajutorul acesteia coloram butoanele in albastru si le punem intr-un array list pt a putea fi selectate.

**4)Metoda MouseClicked**

**->**este o met mostenita care se apeleaza la evenimentul click.La aparitia acestui eveniment butoanele care se afla in arraylist sunt puse pe isVisible(false)

**5)Metoda MouseExit**

**->**este o met mostenita care se apeleaza cand mouseul iese din aria butonului in cadrul acestei metode scoatem butoanele din arraylist si le coloram la culoarea initiala.

**DIAGRAMA UML A POROIECTULUI:**

A screenshot of a cell phone

Description automatically generated