

## Scopul Proiectului

Proiectul "Puzzle Glisant" este o implementare a unui joc de tip puzzle sliding 3x3, unde utilizatorul trebuie să mute piesele pentru a ajunge la o configurație corectă a acestora. Jocul este realizat folosind Java și biblioteca Swing pentru interfața grafică. Piesele sunt amestecate aleatoriu și utilizatorul poate muta piesele adiacente celulei goale pentru a rezolva puzzle-ul.

## Explicația Funcționării Proiectului

### 1. Clasa PuzzleGame:

- **Rol:** Este clasa care gestionează logica jocului. Se ocupă de crearea, amestecarea și verificarea stării tablei de joc.
- **Atribute:**
  - board: Matrice dimensiune 3x3 care reprezintă tabla de joc.
  - emptyRow, emptyCol: Indică poziția celulei goale.
- **Metode:**
  - initializeBoard(): Inițializează tabla cu piesele amestecate și garantează că tabla este rezolvabilă.
  - movePiece(int row, int col): Permite mutarea unei piese în celula goală.
  - isSolved(): Verifică dacă puzzle-ul este rezolvat.
  - isSolvable(): Verifică dacă configurația curentă a tablei este rezolvabilă.
  - shuffle(): Amestecă piesele aleatoriu.

### 2. Clasa Interface:

- **Rol:** Este clasa care se ocupă de interfața grafică a jocului, crează fereastra, butoanele și actualizând tabla de joc pe măsură ce utilizatorul interacționează cu jocul.
- **Atribute:**
  - frame: Fereastra principală a aplicației.
  - buttons: Matrice de butoane care reprezintă piesele jocului.
  - puzzleGame: Instanța clasei PuzzleGame care gestionează logica jocului.
  - boardPanel: Panoul care conține tabla de joc.
  - imagesFolder: Directorul de imagini
- **Metode:**
  - createAndShow(): Crează și afișează interfața grafică.

- `updateBoard()`: Actualizează tabla de joc în funcție de starea curentă a pieselor.
- `loadImageIcon(String filename, int width, int height)`: Încarcă imaginile pieselor.
- `showCustomMessage(String message)`: Afișează un mesaj personalizat într-un dialog atunci când jocul este câștigat.

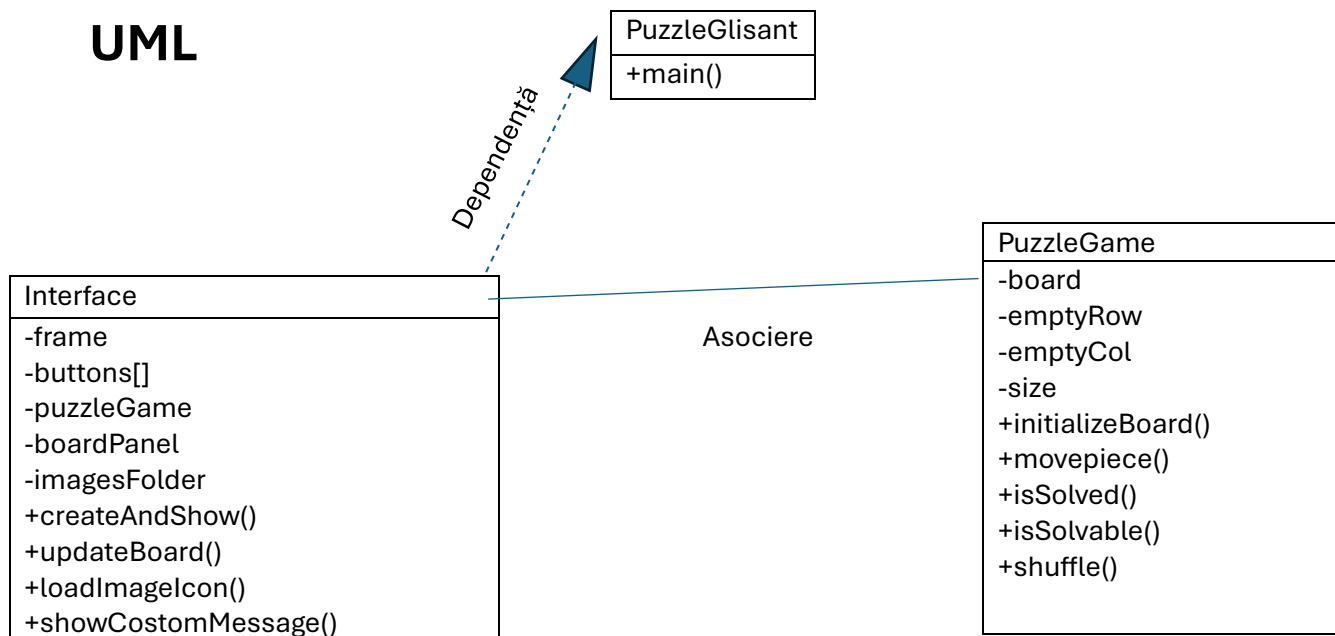
### 3. Clasa **PuzzleGlisant**:

- **Rol:** Este clasa principală care inițiază aplicația și lansează interfața grafică.
- **Metode:**
  - `main(String[] args)`: Lansează aplicația folosind `SwingUtilities.invokeLater` pentru a crea și a arăta interfața grafică.

### Fluxul de Funcționare al Jocului

1. La deschiderea aplicației, se creează un obiect de tip `PuzzleGame`, care generează tabla de joc amestecată.
2. Tabla este reprezentată grafic printr-o rețea de butoane (3x3), fiecare buton având o piesă a puzzle-ului.
3. Utilizatorul poate muta piesele prin apăsarea butoanelor adiacente celulei goale.
4. După fiecare mișcare, se actualizează tabla de joc și se verifică dacă puzzle-ul a fost rezolvat.
5. Dacă puzzle-ul este rezolvat, se afișează un mesaj de felicitare.

## UML



### Explicații ale relațiilor:

Clasa PuzzleGame și clasa Interface:

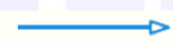
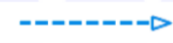
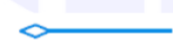
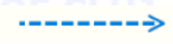


Relația: Asociere.

Motiv: Clasa Interface conține o instanță a clasei PuzzleGame (atributul puzzleGame) pentru a interacționa cu logica jocului.

Clasa Interface și clasa PuzzleGlisant:

Relația: Dependență.

Motiv: Clasa PuzzleGlisant lansează interfața grafică (Interface) în metoda main, folosind SwingUtilities.invokeLater. Relația este de scurtă durată, doar pentru inițializare.

<b>Relație</b>	<b>Simbol</b>	<b>Stil de linie</b>	<b>Forma vârfului</b>
Moștenire		Solid	Triunghi
Implementare de interfață		Întrerupt	Triunghi
Agregare		Solid	Romb
Dependență		Întrerupt	Deschisă
Compozitie		Solid	Romb umplut
Asociere		Solid	