**Proiect EP - Realizarea jocului Sudoku în limbajul C#, folosind algoritmul Forward Checking**

Studenții: Florea Alexandra, grupa 1408A

Pintilie Justinian, grupa 1408A

### **a. Enunțarea temei**

Tema proiectului este rezolvarea jocului Sudoku în limbajul de programare C#, folosind algoritmul Forward Checking. Sudoku este un joc logic popular, în care un grid 9x9 trebuie completat astfel încât:

* Fiecare rând să conțină cifrele de la 1 la 9 fără repetări.
* Fiecare coloană să conțină cifrele de la 1 la 9 fără repetări.
* Fiecare sub-grid 3x3 să conțină cifrele de la 1 la 9 fără repetări.

Această problemă este formulată ca o **problemă de satisfacere a constrângerilor** (CSP - Constraint Satisfaction Problem), ceea ce permite utilizarea unor tehnici avansate de inteligență artificială. Scopul este de a demonstra aplicabilitatea algoritmului Forward Checking în reducerea semnificativă a spațiului de căutare necesar pentru găsirea unei soluții corecte. Algoritmul este implementat în limbajul C#, iar proiectul pune accent atât pe partea funcțională, cât și pe eficiența soluției propuse.

### **b. Arhitectura generală**

Aplicația este structurată în jurul a două componente principale:

1. **SudokuForm**: Aceasta reprezintă interfața cu utilizatorul. Este responsabilă pentru gestionarea interacțiunii utilizator-aplicație, afișarea tabelului Sudoku și evidențierea modificărilor prin culori sugestive (verde pentru mutări valide și roșu pentru mutări invalide).
2. **TableGenerator**: Este clasa care implementează logica jocului și include:
   * Generarea tabelului Sudoku valid folosind algoritmul Forward Checking.
   * Validarea mutărilor utilizatorului.
   * Curățarea tabelei.
   * Generarea sugestiilor/hint-urilor.

Arhitectura aplicată permite separarea clară a responsabilităților, făcând codul ușor de întreținut și extins.

**Fluxul de operare al aplicației**:

* **Generarea tabelei Sudoku**: O tablă completă este generată cu algoritmul Forward Checking. Apoi, se șterg valori pe baza nivelului de dificultate ales (configurat doar de dezvoltator).
* **Interacțiunea utilizatorului**: Utilizatorul introduce valori și primește feedback vizual și mesaje informative.
* **Validare și sugestii**: Funcționalități care sprijină utilizatorul în rezolvarea jocului.

### **c. Funcționalitatea**

Aplicația oferă următoarele funcționalități esențiale:

1. **Generarea unei tabele noi**Funcția TableGenerator.SudokuTableGenerator(int difficulty) este utilizată pentru a genera o tablă validă folosind algoritmul Forward Checking. În funcție de parametrul de dificultate, un număr corespunzător de valori este șters, creând o tablă parțial completată.
2. **Validarea mutărilor utilizatorului**Funcția IsMoveValid(int row, int column, int value) verifică dacă o valoare introdusă de utilizator respectă regulile Sudoku:
   * Valoarea trebuie să fie unică pe rând.
   * Valoarea trebuie să fie unică pe coloană.
   * Valoarea trebuie să fie unică în sub-grila 3x3. Feedback-ul este vizual (verde pentru mutări valide, roșu pentru cele invalide) și textual (mesaje informative despre starea jocului).
3. **Curățarea tabelei**Utilizatorul poate alege să reseteze jocul curent prin curățarea tabelei. Valorile generate inițial rămân neschimbate, însă toate valorile introduse de utilizator sunt eliminate.
4. **Oferirea de hint-uri**Funcția GetPossibleValuesFromForwardChecking(row, column) permite utilizatorului să obțină o sugestie pentru o valoare validă într-o anumită celulă. Numărul de hint-uri disponibile este limitat la 10 și poate fi configurat de dezvoltatori.
5. **Ajutor pentru utilizator**Aplicația include un modul de help care explică regulile jocului Sudoku și descrie principalele funcționalități disponibile.
6. **Feedback vizual și textual**La fiecare acțiune a utilizatorului, aplicația răspunde prin:
   * Evidențierea celulelor valide/invalide.
   * Afișarea de mesaje informative despre progresul jocului, greșelile făcute sau imposibilitatea continuării jocului.

Aceste funcționalități sunt completate de o interfață intuitivă, care oferă utilizatorului o experiență simplă și eficientă.

### **d. Rolul fiecărui membru al echipei**

1. **Florea Alexandra**:
   * Creare proiect pe GitHub și gestionarea acestuia.
   * Implementarea funcțiilor de generare a matricei Sudoku:
     + Algoritmul de generare a tabelelor valide utilizând Forward Checking.
     + Implementarea metodei de validare a mutărilor și a funcției GetHint.
   * Corectarea erorilor întâlnite în timpul dezvoltării.
   * Redactarea următoarelor capitole din documentație:
     + **Capitolul 4**: Modalitatea de rezolvare.
     + **Capitolul 5**: Cod sursă și explicații.
     + **Capitolul 6**: Rezultate obținute.
     + **Capitolul 9**: Contribuții individuale.
2. **Pintilie Justinian**:
   * Crearea scheletului proiectului și configurarea inițială.
   * Implementarea constrângerilor pentru utilizator și a funcției de validare a celulelor.
   * Scrierea primei versiuni a documentației, inclusiv a capitolelor teoretice:
     + **Capitolul 1**: Introducere.
     + **Capitolul 2**: Descrierea problemei.
     + **Capitolul 3**: Aspecte teoretice privind algoritmul Forward Checking.
     + **Capitolul 8**: Bibliografie.

Fiecare membru al echipei a contribuit esențial la realizarea proiectului, combinând cunoștințele tehnice cu abilitățile de documentare pentru a asigura un rezultat final de calitate.