



Sudoku Applikation

Shiva Moradi
2024.02.06

Jag valde att lösa Sudoku-pusselen. I den här beskrivningen kommer jag att berätta och analysera min arbete.

Målet med projektet är att lösa Sudoku pusselen med Python och NumPy. Appen kommer att ta input (gissa siffran) från användare och visa outputet.

Sudoku reglerna:

1. Varje rad måste innehålla alla siffror från 1 till 9.
2. Varje kolumn måste innehålla alla siffror från 1 till 9.
3. Varje 3x3-underrutnät måste innehålla alla siffror från 1 till 9.

Man ska fylla varje rad, varje kolumn och varje 3x3-underrutnät med siffror från 1 till 9 så att varje siffra endast förekommer en gång i varje rad, kolumn och 3x3-underrutnät. Pusslet börjar med några fördefinierade siffror som kallade "given" och dem redan är placerade på brädet.

Följande steget kommer jag ta under projektet:

- Beskriva reglerna som kommentar och börja skriva Python funktioner utifrån kommenterade reglerna.
- En användare kommer att börja och slutföra pusslet genom att gissa siffran (mellan 1-9) som följer pussle reglerna.
- Användaren ska börja med att titta på de siffror som redan är placerade på brädet. Dessa leder användaren för att fylla i tom ruta med möjliga siffror.
- Min app börjar att placera siffran från första tom cell i första rad och går vidare rad till rad.

Hur fungerar min applikation?

```
given = [  
    [5, 3, 0, 0, 7, 0, 0, 0, 0],  
    [6, 0, 0, 1, 9, 5, 0, 0, 0],  
    [0, 9, 8, 0, 0, 0, 0, 6, 0],  
    [8, 0, 0, 0, 6, 0, 0, 0, 3],  
    [4, 0, 0, 8, 0, 3, 0, 0, 1],  
    [7, 0, 0, 0, 2, 0, 0, 0, 6],  
    [0, 6, 0, 0, 0, 0, 2, 8, 0],  
    [0, 0, 0, 4, 1, 9, 0, 0, 5],  
    [0, 0, 0, 0, 8, 0, 0, 7, 9]  
]
```

Jag har börjat med att importera NumPy för att konvertera den tvådimensionella listan till en NumPy-matris och tilldelar den till variabeln `ngiven`.

Nollar i min 9x9 Sudoku-matris, representerar tomma celler, och tilldelar den till variabeln `given`.

Sedan skrivit ut den initiala matrisen för att visualisera det tomma Sudoku-pusslet.

`prettyfy()` Funktionen:

Funktionen skriver ut Sudoku-matrisen på ett snyggt sätt med rader och kolumner tydligt markerade. Använder loopar för att gå igenom varje rad och kolumn och infogar rätt separatorer och avgränsare för att göra det mer lättläst.

`board(row, column)` Funktionen:

Funktionen `board(row, column)` låter användaren gissa och fylla i en siffra på en given rad och kolumn. Kontrollerar om gissningen är korrekt enligt Sudoku-reglerna (ingen repetition i rad, kolumn och 3x3-rutan). Om gissningen är korrekt, uppdateras Sudoku-matrisen, och funktionen returnerar `True`, annars `False`.

solve(given) Funktionen:

Funktionen `solve(given)`, använder backtracking för att lösa Sudoku-pusslet. Loopar genom varje cell i matrisen och försöker fylla i en siffra med `board()`-funktionen. Om en siffra är giltig och leder till en lösning, fortsätter rekursivt processen. Om ingen giltig siffra kan fyllas in, återställs cellen och backtracking fortsätter.

Slutligen, anropar `solve(given)` för att lösa Sudoku-pusslet med användarinteraktion och även anropar `prettify()` för att visa det lösta pusslet.

Länk till GitHub: <https://github.com/ShivaMoradi/Sudoku>