

Project report:

GUI-Based Sudoku solver

-Nischay Gothwal
-Sumadura Upparahal

LINK: <https://github.com/sumadura/Sudoku-Solver.git>

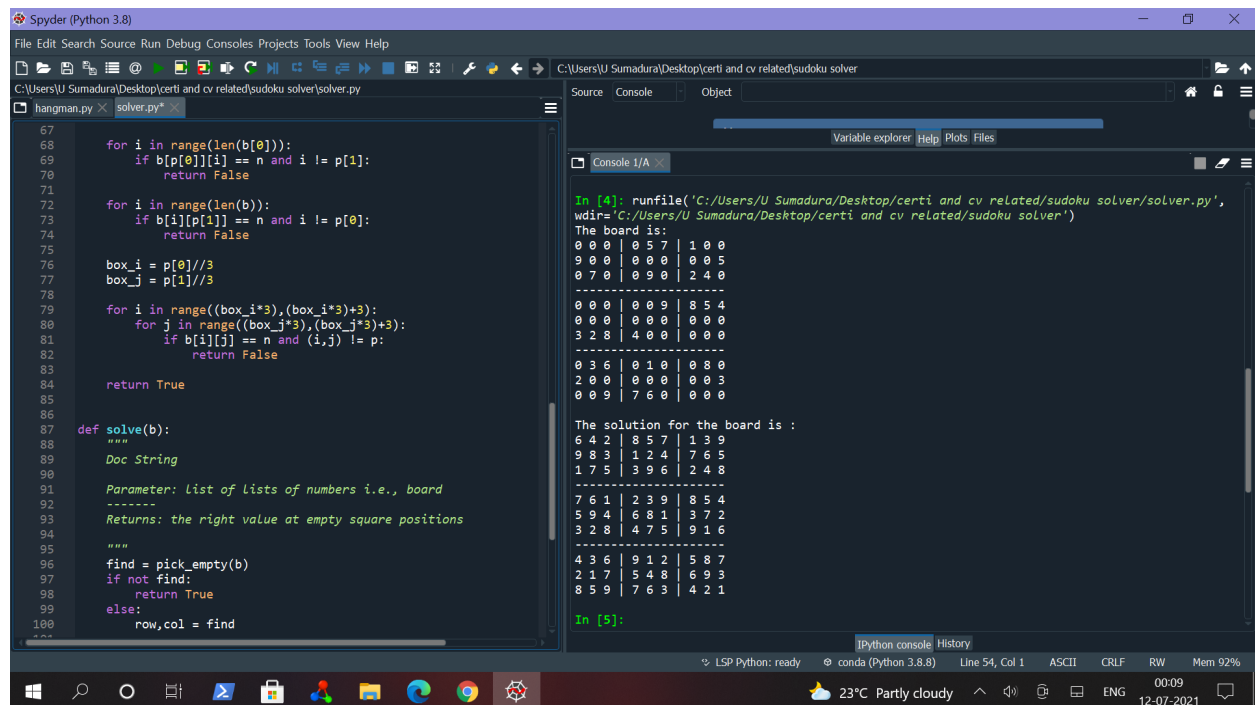
1. Statement:

This code helps visualize the backtracking algorithm and solves any sudoku puzzle using Python and Pygame. The repository contains 2 python codes:

- **Sudoku_GUI.py:** This code is a visualization tool as well as a working sudoku game in which the code tells us if the user input is valid or not, it will also add the number if it is valid on that location.
- **solver.py:** This code is just a sudoku solver without 'backtracking algorithm visualization' to solve the sudoku puzzle faster.

NOTE - This code only works for solvable sudoku puzzles.

2. Relevant screenshots:



The screenshot displays the Spyder Python IDE interface. The left pane shows the source code for `solver.py`, which includes a `valid` function for checking Sudoku constraints and a `solve` function using backtracking. The right pane shows the console output where the `solve` function is executed on a specific 9x9 board. The output displays the input board, followed by the solved board.

```
In [4]: runfile('C:/Users/U Sumadura/Desktop/certi and cv related/sudoku solver/solver.py',
wdir='C:/Users/U Sumadura/Desktop/certi and cv related/sudoku solver')
The board is:
0 0 0 | 0 5 7 | 1 0 0
9 0 0 | 0 0 0 | 0 0 5
0 7 0 | 0 9 0 | 2 4 0
-----
0 0 0 | 0 0 9 | 8 5 4
0 0 0 | 0 0 0 | 0 0 0
3 2 8 | 4 0 0 | 0 0 0
-----
0 3 6 | 0 1 0 | 0 8 0
2 0 0 | 0 0 0 | 0 0 3
0 0 9 | 7 6 0 | 0 0 0

The solution for the board is :
6 4 2 | 8 5 7 | 1 3 9
9 8 3 | 1 2 4 | 7 6 5
1 7 5 | 3 9 6 | 2 4 8
-----
7 6 1 | 2 3 9 | 8 5 4
5 9 4 | 6 8 1 | 3 7 2
3 2 8 | 4 7 5 | 9 1 6
-----
4 3 6 | 9 1 2 | 5 8 7
2 1 7 | 5 4 8 | 6 9 3
8 5 9 | 7 6 3 | 4 2 1

In [5]:
```


Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\U Sumadura\Desktop\certi and cv related\sudoku solver\Sudoku_GUI.py

```
7 class Grid:
8     board = [
9         [0, 0, 0, 2, 6, 0, 7, 0, 1],
10        [6, 8, 0, 0, 7, 0, 0, 9, 0],
11        [1, 9, 0, 0, 0, 4, 5, 0, 0],
12        [8, 2, 0, 1, 0, 0, 0, 4, 0],
13        [0, 0, 4, 6, 0, 2, 9, 0, 0],
14        [0, 5, 0, 0, 0, 3, 0, 2, 8],
15        [0, 0, 9, 3, 0, 0, 0, 7, 4],
16        [0, 4, 0, 0, 5, 0, 0, 3, 6],
17        [7, 0, 3, 0, 1, 8, 0, 0, 0]
18    ]
19
20
21    def __init__(self, rows, cols, width, height):
22        self.rows = rows
23        self.cols = cols
24        self.cubes = [[Cube(self.board[i][j]) for j in range(self.cols)] for i in range(self.rows)]
25        self.width = width
26        self.height = height
27        self.model = None
28        self.update_model()
29        self.selected = None
30        self.win = win
31
32
33    def update_model(self):
34        self.model = [[self.cubes[i][j].value for j in range(self.cols)] for i in range(self.rows)]
35
36
37    def place(self, val):
38        """
39        Doc String
40        """
```

Sudoku

4	3	5	2	6	9	7	8	1
6	8	2	5	7	1	4	9	3
1	9	7	8	3	4	5	6	2
8	2	6	1	9	5	3	4	7
3	7	4	6	8	2	9	1	5
9	5	1	7	4	3	6	2	8
5	1	9	3	2	6	8	7	4
2	4	8	9	5	7	1	3	6
7	6	3	4	1	8	2	5	9

Time: 0:0:13

Console History

LSP Python: ready | conda (Python 3.8.8) | Line 25, Col 27 | ASCII | CRLF | RW | Mem 91%

23°C AQI 84 | 00:20 | 12-07-2021