**AI PROJECT**

**SUDOKU GAME DESIGN FOR 8 x 8 GRID**

By

Konakalla Naga Venkata Nikhil - AP18110010071

Are V S S Tharun - AP18110010074

Kanulla Sai Krishna - AP18110010080

Chaitanya Palakayala - AP18110010083

Tadekoru Sri Krishna - AP18110010084

Tumula Mani Harsha - AP18110010088

Velaga Sai Surya - AP18110010090

Pattan Mehafooz Ali Khan - AP18110010092

**Abstract**

Solving sudoku puzzle is a passion. Although looking easy it carries much more concentration.

The simplicity of puzzle’s structure and the low requirement of mathematical skills causes interest in solving it.

Sudoku Puzzles can be solved using many techniques like back tracking, Stochastic search/optimization methods, Constraint programming, Exact cover, Relations and residuals.

In this project the solving of sudoku puzzle is done using the back tracking technique.

**Introduction**

Sudoku is logical puzzle presented on a square grid that is usually 8X8, 9X9, 16X16 or other sizes. In this document we will consider only 8X8 case although almost everything that is said can easily be extended to puzzles with different dimensions.

The aim of the puzzle is to enter a numerical digit from 1 to 8 in each cell of a 8X8 made up of 2X4 sub-rectangles, at the start only a few numbers are revealed. each row, column and sub-rectangles region must contain each of the number 1 to 8 exactly once. Accordingly, we have use logic to deduce the position and order of the rest of the numbers in the entire grid.

As the problem here is solved using the back tracking method, Each cell is tested for a valid number, moving "back" when there is a violation, and moving forward again until the puzzle is solved.

Advantages of using back tracking

• A solution is guaranteed (as long as the puzzle is valid).

• Solving time is mostly unrelated to degree of difficulty.

• The algorithm (and therefore the program code) is simpler than other algorithms, especially compared to strong algorithms that ensure a solution to the most difficult puzzles.

**Rules**

* Every square (cell) has to contain a single number
* Only the numbers from 1 through to 8 can be used
* Each 2x4 box can only contain each number from 1 to 8 once
* Each vertical column can only contain each number from 1 to 8 once
* Each horizontal row can only contain each number from 1 to 8 once

Once the puzzle is solved, this means that every row, column, and 2x4 box will contain every number from 1 to 9 exactly once. In other words, no number can be repeated in any 2x4 box, row, or column. And sum of the elements in a row or column or 2x4 box is same.

**Methodology**

The project is developed with the backtracking algorithm using Graphical User Interface in Python. In the beginning the sudoku board is initialized with some 8x8 grid. And the project contains several methods. They are: -

**draw(): -**

This method, draws the required lines in a sudoku of 8x8.

**draw\_box(): -**

This method, actually highlights the cell that we select at run time.

**solve(): -**

In the method, the entire backtracking algorithm is done. This is where the program starts. It is a recursive function. If some value in a cell didn’t followed the rules, it backtracks and tries another value.

**find\_empty(): -**

Through the method, we get the co-ordinates of an empty cell.

**valid(): -**

This method, checks whether a certain value can be inserted into a particular empty cell. It makes sure that the value should follow the rules, specified in rules section.

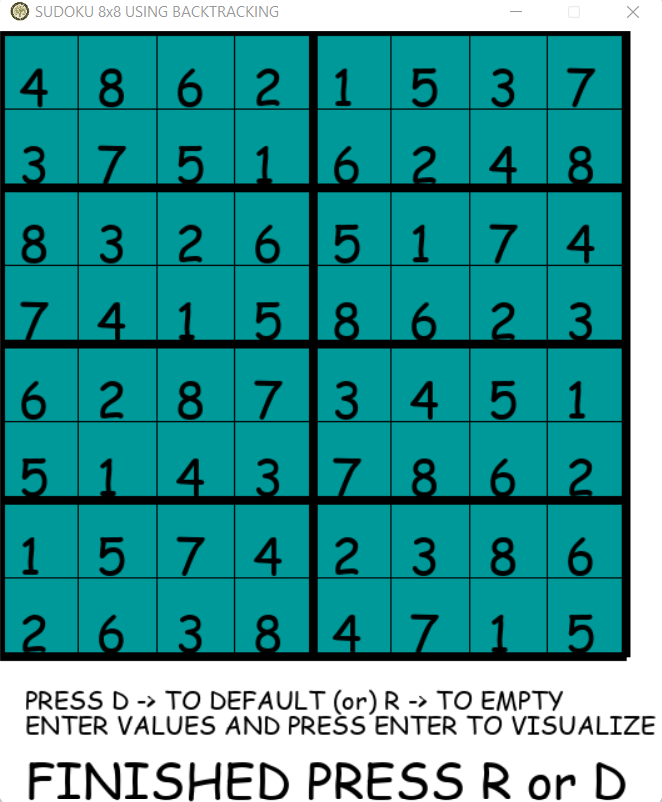
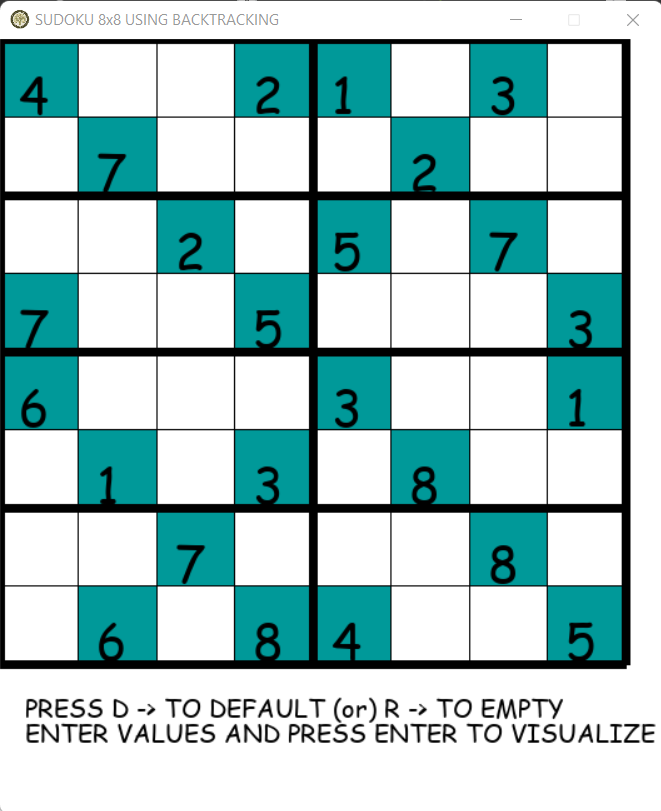
**Requirements**

Operating System: Windows, mac

Storage: minimum 1 GB

Software Requirements: Python, Jupyter notebook, Pygame library.

**Results**



**References**

1. <https://realpython.com/pygame-a-primer/>

2. <https://www.geeksforgeeks.org/sudoku-backtracking-7/>

3. <https://www.geeksforgeeks.org/building-and-visualizing-sudoku-game-using-pygame/>