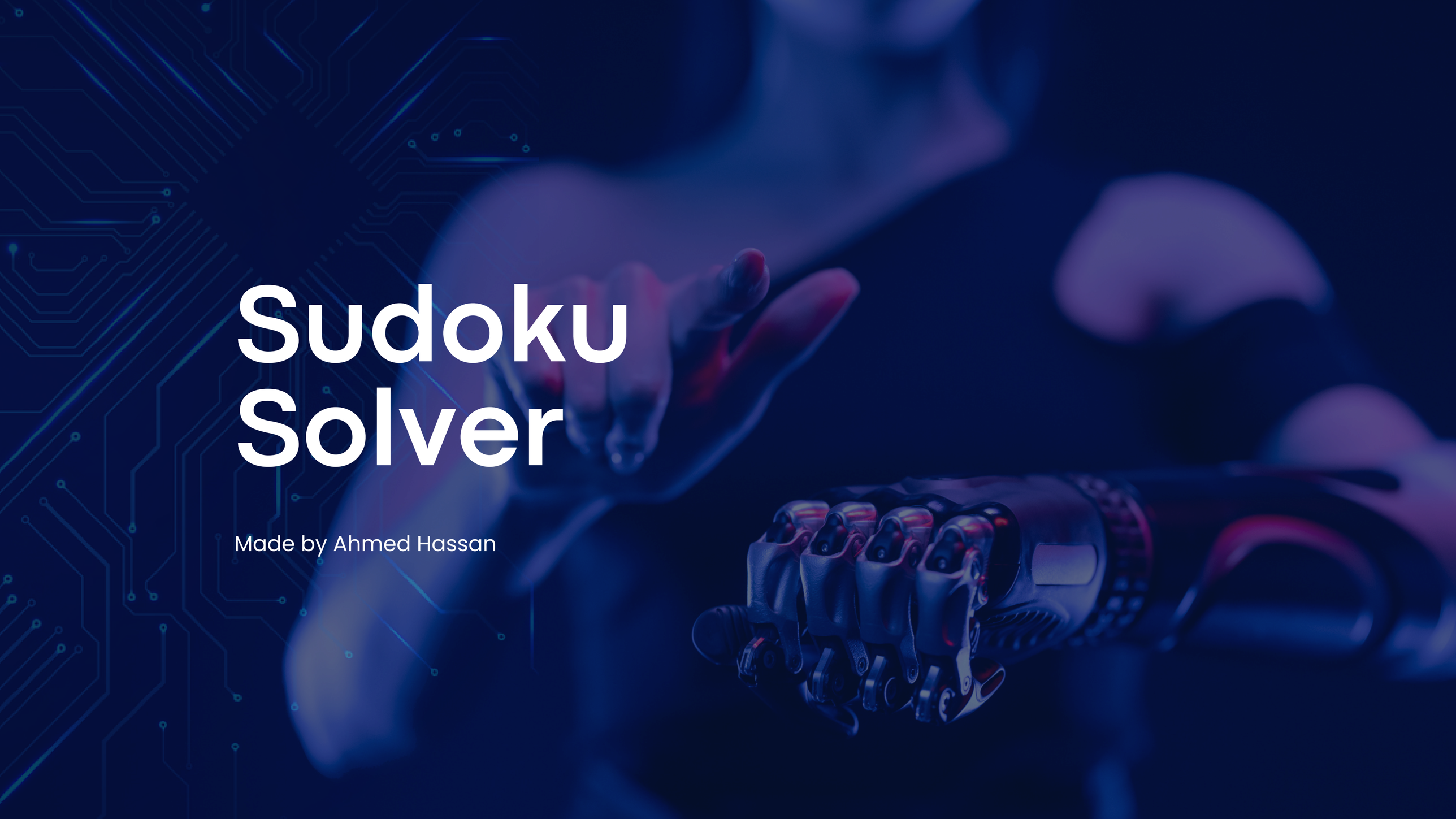
Sudoku Solver

Made by Ahmed Hassan



The Goal

- Sudoku is a puzzle, where The objective is to fill a 9×9 grid so that each column, row, and the nine 3×3 box contains all digits from 1 to 9.
- We want to make an AI that can solve Sudoku Puzzles using blind(uninformed) search.

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

The Environment

Fully Observable

he agent can see the entire board

Determanistic

the outcome of placing a number is predictable

Static

only the agent’s actions change the board

Descrete

There is a finite number of cells and possible values

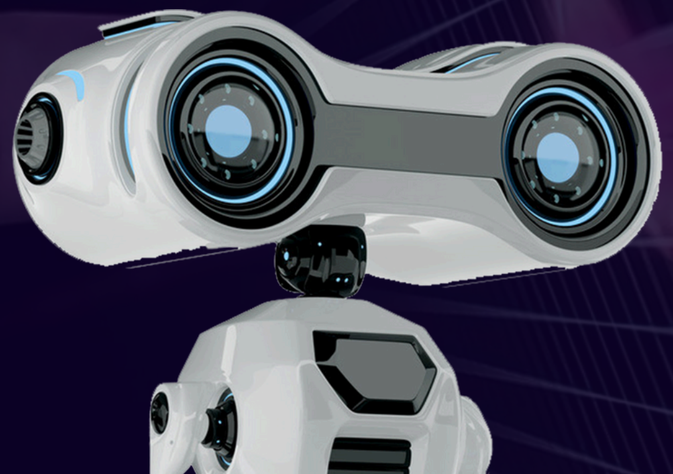
The Approach

our approach to this problem is made of four main steps:

1. Get a well-designed puzzle directly from sudoku.com
2. Determine what numbers can be placed in each empty cell
3. Solve using backtracking (blind search)
4. Show everything on a GUI

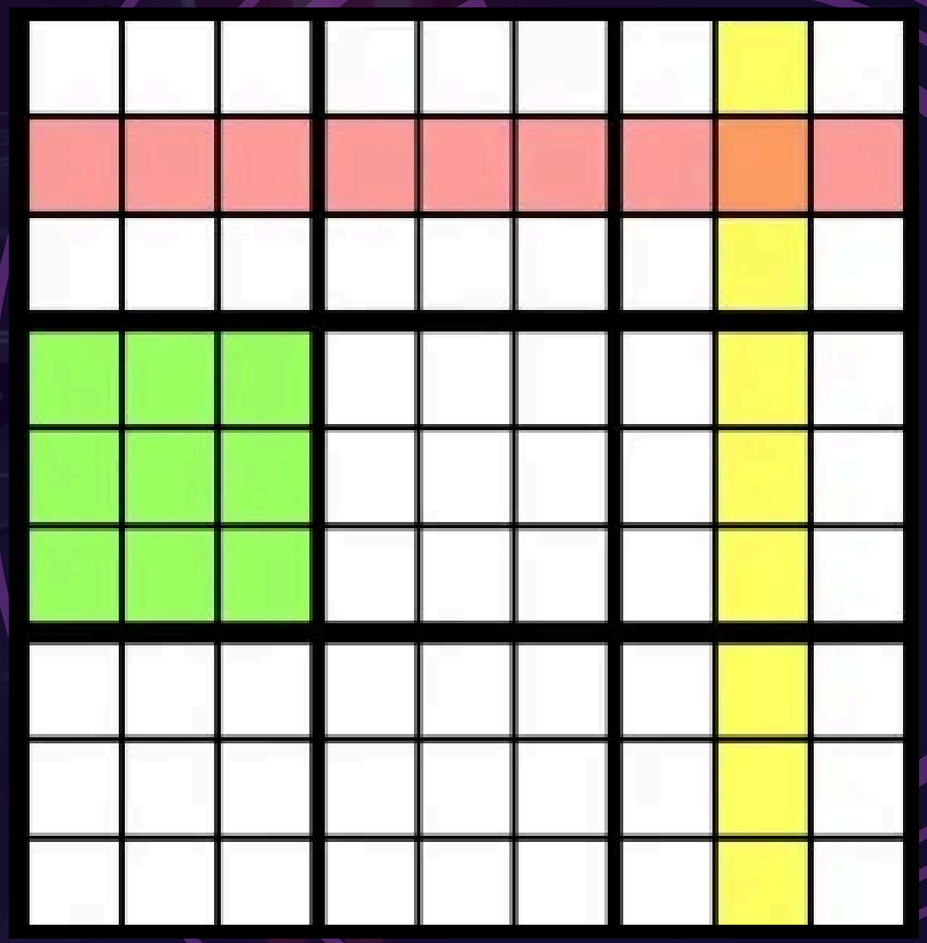
1. Getting the puzzle

- It is a simple but important part of the project
- we call the sudoku.com API to get a puzzle for the agent to solve
- doing this insures having a well-designed puzzle that has a solution (only one)



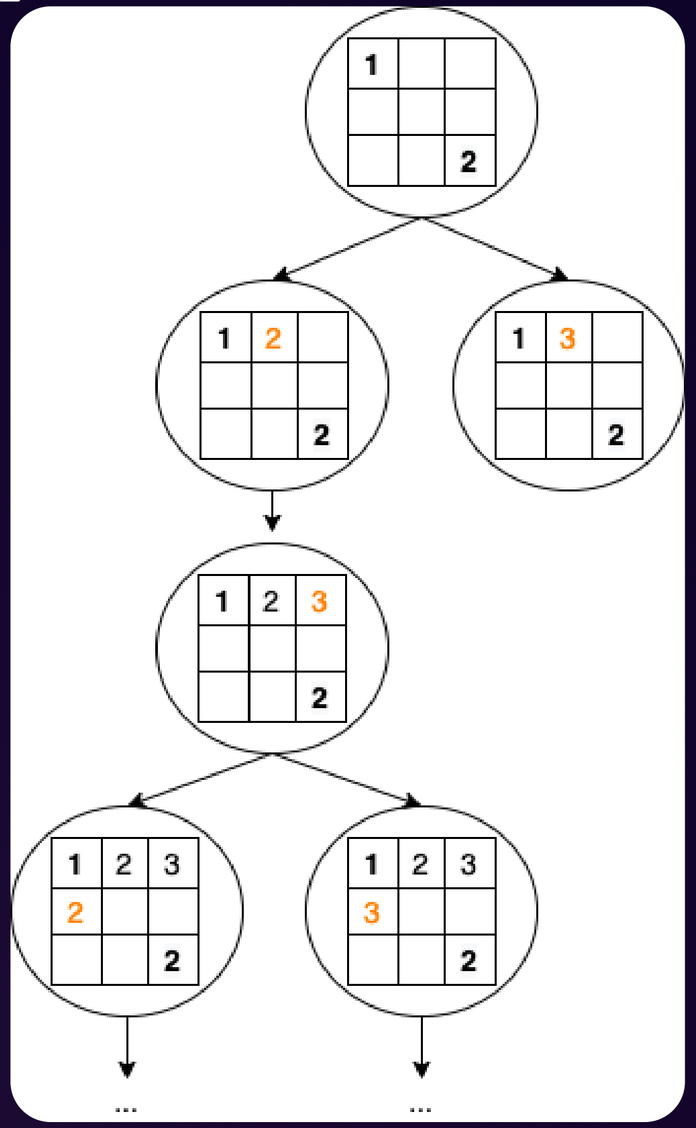
2. Knowing the Possibilities

- First we iterate on each cell in the 9x9 grid, searching for empty cells
- If a cell is empty we use simple sudoku rules to know what numbers can be placed in it (1, 2, 4, ..)
- continue until we complete every cell
- All the possibilities will be placed in one place to be accessed in the next step



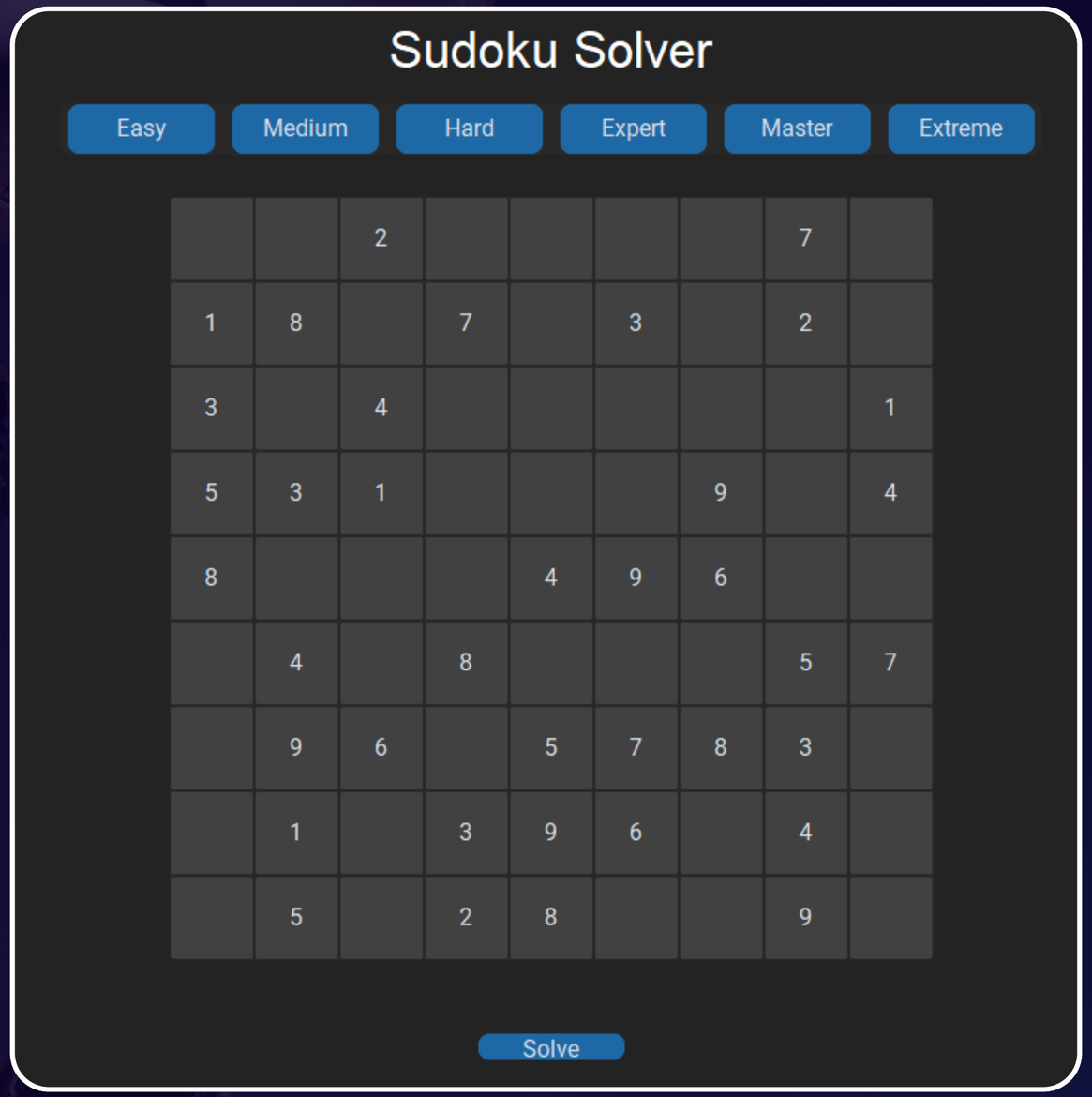
3. BackTracking(DFS)

- To apply our blind search approach we implemented Depth-First Search
- DFS uses the possibilities determined in the previous step and tries all of their combinations
- It only uses numbers that are possible in the current state
- If it gets stuck (no number in this cell's possibilities can be placed here) it trackbacks to try another branch



4. GUI

- We made a simple GUI that
- It shows trackback in action and the puzzle selected
- we can also select the difficulty level



Approach pros and cons

Pros	Cons
Easy to implement	Blind
Relatively Memory efficient	Slow (expert and extreme diff)
Always reaches the Goal	Not optimal for sudoku


Thank You!