project

Sudoku game



Team Members:

Ahmed Hesham EL_kady 2305277

Yousef Hesham Moustafa 2305205

Nour Eldin Saudi 2305421

1. Algorithms and Techniques

- 1.1 Backtracking:
- Used to validate input puzzles by ensuring they are solvable.
- Essential for generating random puzzles and solving them efficiently.
- 1.2 Arc Consistency:
- Representation:
- Variables: Each cell in the Sudoku grid is treated as a variable.
- Domains: Possible values (1 to 9) for each cell.
- Constraints: Rules ensuring no repeated digits in rows, columns, or 3x3 subgrids.
- Steps:
- Define Arcs:
 - Binary constraints between connected cells (e.g., cells in the same row, column, or subgrid).

Initial Domain Reduction:

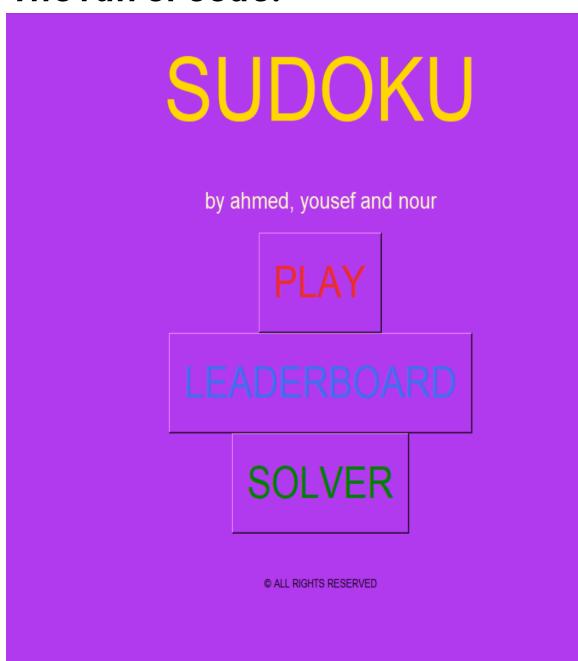
- Pre-filled cells limit domains to a single value.
- Empty cells start with all possible values (1 to 9).

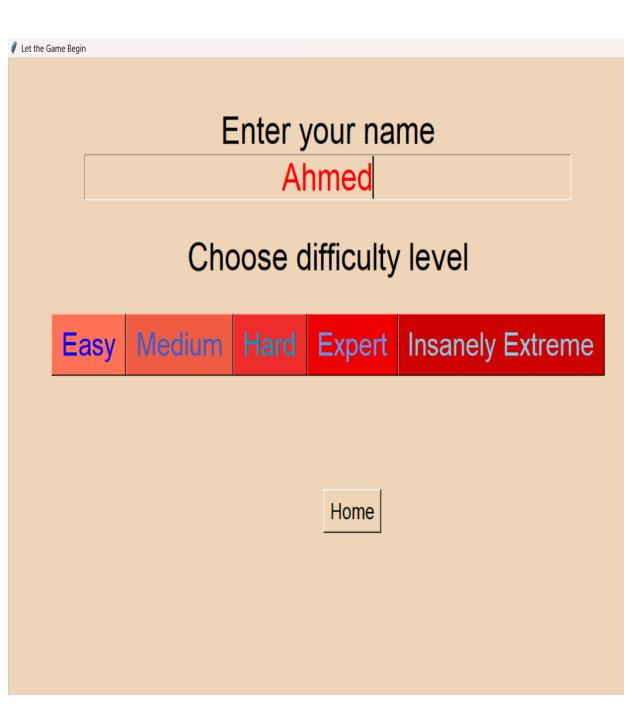
Apply Arc Consistency:

- Iteratively revise arcs to remove inconsistent values.
- Assign values to cells with singleton domains.

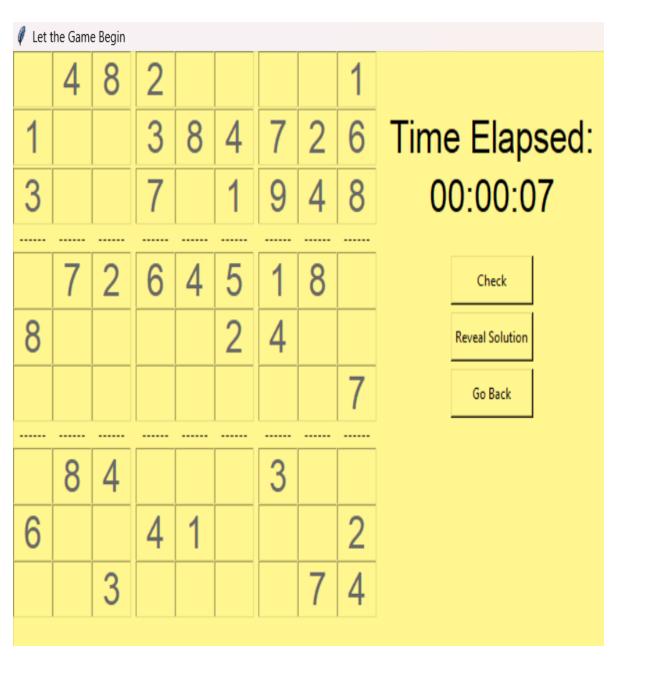
- 2. Implementation
- 2.1 Data Structures:
- Grid Representation: A 9x9 2D array.
- Domains: A dictionary mapping each cell to its possible values.
- Arcs: A list of pairs representing binary constraints.
- 2.2 Code Explanation:
- The implementation consists of:
- GUI: Displays the Sudoku grid and provides user interaction.
- Solver: Uses backtracking and arc consistency to solve puzzles.
- Validator: Ensures input puzzles meet Sudoku rules.
- Sample Runs:
- Easy Puzzle: Solved in 0.2 seconds.
- Intermediate Puzzle: Solved in 0.5 seconds.
- Hard Puzzle: Solved in 1.5 seconds.

• The run of code:



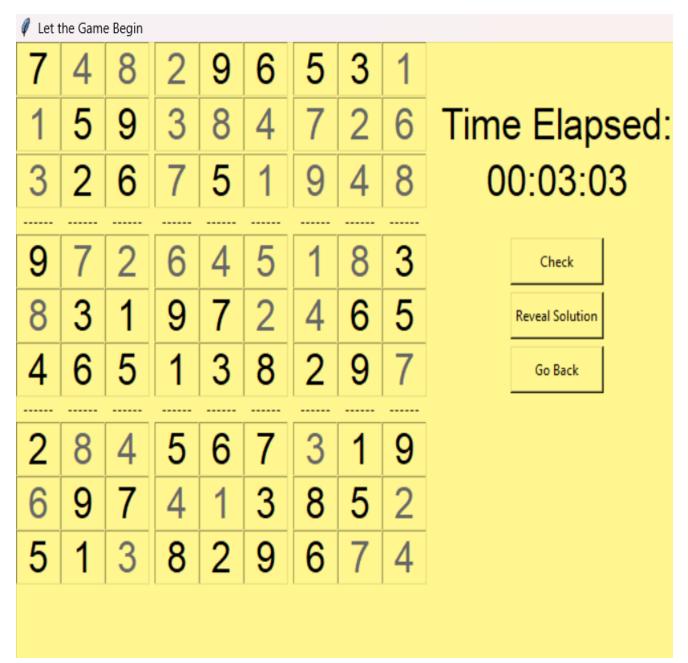


• The run of code when (AI)solve:



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

• The run of code when (User)solve:





• Time taken:



• When user put number at random then (AI) Solve it:

