Animated Sudoku Solver - Project Report

# 1. Introduction

This project implements an Animated Sudoku Solver using Python, Pygame for GUI, and Google OR-Tools for solving the puzzle using constraint programming. The user inputs a Sudoku puzzle and upon pressing the 'Solve' button, the solver visually animates the solution process.

# 2. Technologies Used

• Python 3  
• Pygame  
• Google OR-Tools (CP-SAT Solver)  
• Object-Oriented Programming

# 3. Key Features

• Visually appealing GUI using Pygame  
• Real-time animation of the solving process  
• Constraint programming approach to solve Sudoku  
• Efficient and accurate solution generation

# 4. Working Mechanism

1. The GUI initializes a Sudoku board with some pre-filled values.  
2. The user clicks the 'Solve Puzzle' button to start the solving process.  
3. The solver uses constraint programming to fill in the board ensuring that each row, column, and 3x3 box contains all digits from 1 to 9 without repetition.  
4. The animation visually displays the insertion of each value in the grid.  
5. Once the puzzle is solved, the GUI displays a success message.

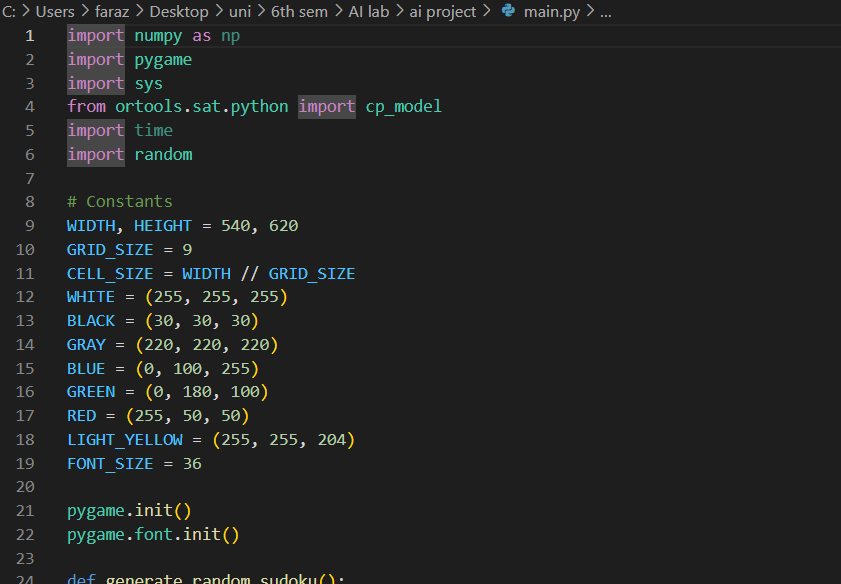
# 5. Sudoku Constraints

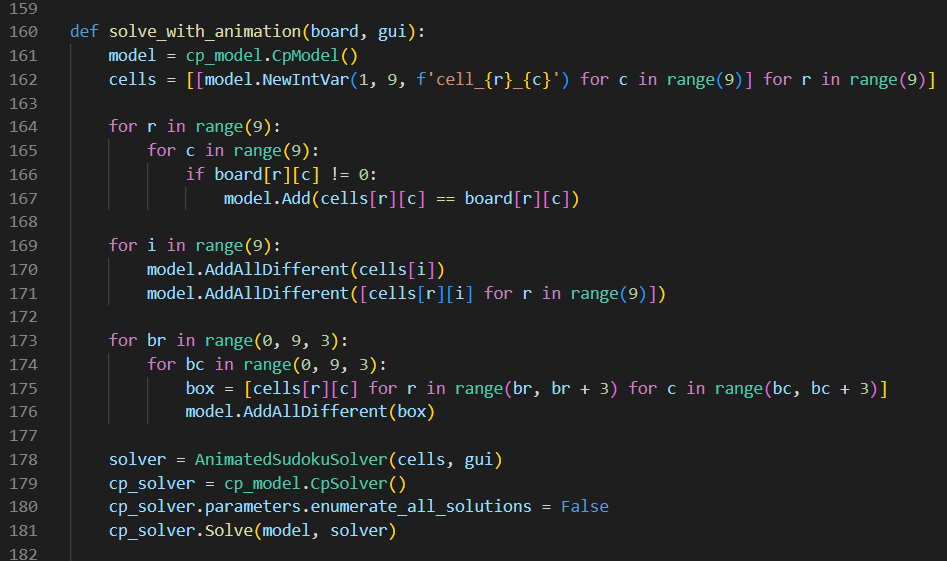
• Each cell is a variable with values from 1 to 9.  
• Pre-filled cells are fixed using equality constraints.  
• Each row must contain distinct digits (AddAllDifferent).  
• Each column must contain distinct digits (AddAllDifferent).  
• Each 3x3 box must contain distinct digits (AddAllDifferent).

# 6. Conclusion

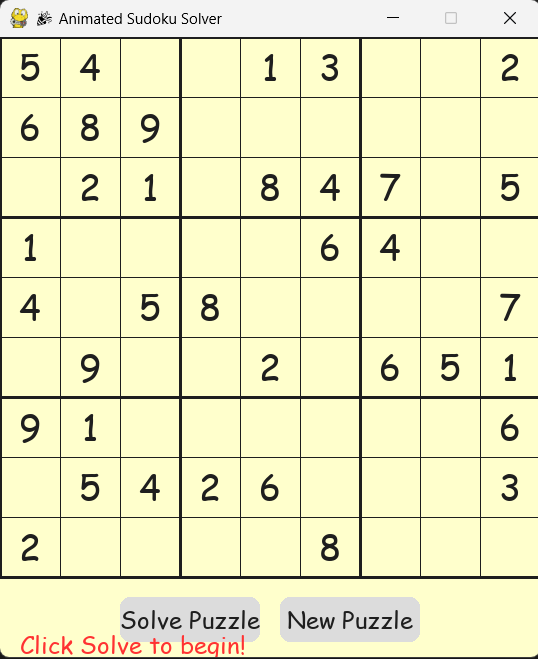
This project successfully demonstrates the integration of graphical user interfaces with constraint-based artificial intelligence to solve combinatorial problems like Sudoku. The animation enhances the user experience and helps in understanding the step-by-step process of solving.

Pictures:

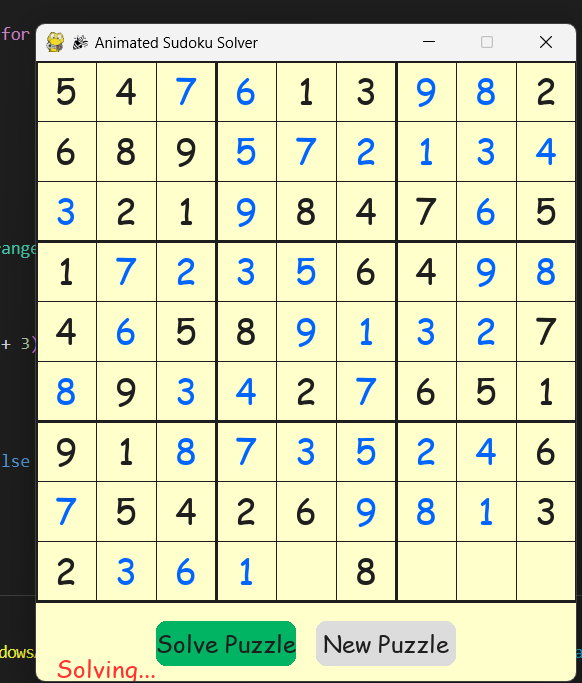




Initial board with GUI:



While solving Puzzle:



Generating a new puzzle:

