

Project Title: Sudoku Solvers

Team Members:

- Michael Richards - mrichards2021@my.fit.edu
- Jaden Krekow - jkrekow2021@my.fit.edu
- Alice Luce - pluce2021@my.fit.edu
- Adrian Rodriguez - adrianrodrig2019@my.fit.edu

Faculty Advisor: Raghuveer Mohan - rmohan@fit.edu

Client: Not Applicable

Dates of Meetings with the Client for developing this Plan: Not Applicable

Goal and Motivation:

The goal of sudoku solvers is to create an app that allows users to generate sudoku puzzles at the press of a button. Not only will it generate puzzles but also give users an option to solve the puzzle if they wish to see what the finished result will look like. Current systems that implement a way to play sudoku suffer from limited variety, abundant ads, and poor user interface. These issues cause frustration among potential users and can deter them from using those versions of sudoku or make the users lose interest in it.

Approach (key features of the system):

Generate puzzles

The user can create a new randomly generated board with a simple button click. This will make sure you never play the same board with countless possible board generations.

Auto solver

While the user is solving the sudoku puzzle, they will have access to a button that will instantly solve the puzzle for them. This auto solver will use a modern algorithm or artificial intelligence techniques to solve the puzzle in a reasonably fast amount of time. This auto solver will be useful to the user if they ever get stuck when trying to solve the puzzle. We plan to expand upon this feature and add a "hint" button that can give the user the correct number for a cell that the user is having difficulty filling in.

Having trouble finishing the puzzle? Or just curious to see what it would look like? You will be able to have our application automatically complete any puzzle with a press of a button. This process will fill in each spot in the puzzle with the correct number, showing you how it would look when it's finished.

Various difficulties

Our application will be able to generate a random sudoku puzzle with exactly one solution upon the user starting a new game. This puzzle will be a level of difficulty depending on which difficulty setting the user picks. The difficulty of the puzzle will be measured by how much information the puzzle gives to the user. The less information in the puzzle, the harder the difficulty will be and vice versa. The amount of information can be measured by how many cells are already filled in the puzzle.

The user can challenge themselves by asking the system to generate sudoku puzzles to start with a different number of spaces already filled in. The user can increase or decrease the number to match their preferences.

Gui Application

The User Interface Application will be the way the user can interact with the sudoku puzzle. They will be able to launch the application and be greeted with a menu screen where they can resume their last game or start a new game of sudoku with a variety of difficulty settings for the user to choose from. They will then be launched into the game itself and be able to interact with the empty cells in the sudoku puzzle to try and solve it. During their gameplay, they will have access to a button that will instantly solve the puzzle for the user if they get stuck and would like to see the solution.

The user can navigate easily through the application with simplistic approach emphasizing easily understandable buttons that allow the user to go where they want with minimal effort

Unique Solution

The user can be sure that when they start a puzzle it will be unique, ensuring that there can be only one way the generated puzzle can be completed.

Novel Features/Functionalities:

Sudoku Solver/ Generator is not very novel, but we will make a novel GUI through controls to easily navigate cells in the puzzle and a good looking, user friendly interface.

Algorithms and Tools:

PyQT for the GUI interface. Backtracking algorithms to solve puzzles.

Technical Challenges:

1. Before we can start our main task of creating an auto solver we have to create the base game and GUI menu
2. When generating sudoku puzzles, it will be a challenge to make sure the puzzle has only one unique solution (Approximately 6.671×10^{21} unique Sudoku boards)

3. We are not well versed in different Sudoku solving algorithms so testing and research would need to be done to determine efficiency between different approaches

Milestone 1 (Sep 30) itemized tasks:

1. Research efficient solving strategies and unique game generation for Sudoku
2. Compare and select GUI libraries within python and collaboration tools for code and work documents
3. Create Requirement Document
4. Create Design Document
5. Create Test Plan

Milestone 2 (Oct 28) itemized tasks:

1. Determine data structure to store Sudoku game board
2. Make GUI Interface with blank Sudoku game board that can be edited
3. Implement, test, and demo a base Sudoku game

Milestone 3 (Nov 25) itemized tasks:

1. Determine technique to test Sudoku board has one solution
2. Implement, test, and demo a Sudoku generator that generates games with a unique solution

Task matrix for Milestone 1:

Task	Michael	Jaden	Alice	Adrian
Research efficient solving strategies and unique game generation for Sudoku	Research 25%	Research 25%	Research 25%	Research 25%
Compare and select GUI libraries within python, collaboration tools for code and work documents	Research 25%	Research 25%	Research 25%	Research 25%
Create Requirement Document	Write 25%	Write 25%	Write 25%	Write 25%
Create Design Document	Write 25%	Write 25%	Write 25%	Write 25%
Test Plan	Test & Debug 25%	Test & Debug 25%	Test & Debug 25%	Test & Debug 25%

Approval from Faculty Advisor:

"I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____