Sudoku Solver Application

Overview

The Sudoku Solver Application is a Qt-based graphical user interface that allows users to play and solve Sudoku puzzles. The application includes a Sudoku board, a solver algorithm, and interactive features to enhance the user experience.

Table of Contents

Features

- Interactive Sudoku board with clickable cells.
- Sudoku solver algorithm to automatically complete puzzles.
- User-friendly interface with visual cues for better usability.
- Dynamic UI improvements for an enhanced user experience.

Getting Started

Prerequisites

- Qt framework installed on your system.
- C++ compiler compatible with the Qt version.

Installation

1. Clone the repository:
   git clone https://github.com/yourusername/SudokuSolver.git
2. Build the application using Qt Creator or your preferred Qt development environment.
3. Run the application.

How to Use

User Interface

The application provides an intuitive user interface with the following components:

- **Sudoku Board:** A 9x9 grid where users can input their Sudoku puzzle or play interactively.
- **Interactive Controls:** Buttons to validate, solve, and reset the Sudoku board.
- **Status Display:** Information about the solving process and results.

Game Rules

Sudoku is a number puzzle game played on a 9x9 grid. The rules are simple:

1. Each row must contain the numbers 1 through 9 with no repetition.
2. Each column must contain the numbers 1 through 9 with no repetition.
3. Each of the nine 3x3 subgrids must contain the numbers 1 through 9 with no repetition.

Solver Algorithm

The application includes a backtracking algorithm to solve Sudoku puzzles. The solver checks each cell for a valid number based on the game rules, and if a solution exists, it fills in the entire board.

UI Improvements

To enhance the user experience, the application includes the following UI improvements:

- **Cell Highlighting:** Selected cells are highlighted in medium blue for easy identification.
- **Error Feedback:** Cells with incorrect values are highlighted in red to alert the user.
- **Dynamic Background Colors:** Cells are visually categorized based on their status (empty, filled, error) to improve clarity.

License

Acknowledgments

I would like say thank you to Scalian and 42 Malaga Fundacion Telefonica for giving me the opportunity to participate in this engaging code challenge. The experience has been valuable, and I appreciate the chance to showcase my skills.

Despite me being new to the QT framework, the learning experience was enjoyable, and I am grateful for the chance to explore and work with a new technology.

Thank you for the enriching experience!

made with ❤️ by Sol Andrade