

Sudoku in C++

by

Connor Gallagher

Overview

Sudoku in C++ was created by Connor Gallagher as a final project for Dr. Butcher's Open Source Software class. It is licensed under the MIT License and all source code for this project can be found here:

github.com/cgallagher9/Comp312.

The game is written entirely in the C++ programming language, runs in the command prompt, and reads keyboard interactions from the user.

Thank you for downloading, and I hope you enjoy playing.

Installing and Running

In this package, you will find the following:

- Sudoku.exe
- Boards Folder
- Saved_Boards Folder
- Solutions Folder
- Sources Folder

To play Sudoku_Game.exe, simply double-click the executable file's icon: no installation necessary.

NOTE: Although it is possible to successfully play Sudoku Game with the executable removed from its original folder, this is NOT recommended. The code is designed to look for text files in the Boards, Solutions, and Saved_Boards folders. If Sudoku_Game.exe is removed from its original location, the program will still function. However, each time it is run, the program will ask the user for the locations of both the sudoku board text files and the solution board text files. It is then up to the user to copy the correct file paths and paste them into the command prompt.

Alternatively, if the Boards, Saved_Boards, and Solutions folders are in static locations, you can change the following variables' values in the source code to automate the game's lookups of the text files.

```
75| string inFileName = "/*Path to Boards folder*/"+boardName+".txt";  
76| string solution_path = "/*Path to Solutions folder*/"+boardName+".txt";  
191| string outFileName =  
    "/*Path to Saved_Boards folder*/"+boardName+".txt";
```

Building From Source Code

To build the game yourself, simply import the files from the Sources folder into

Game Instructions

The player must insert a number between 1 and 9 in each cell on the sudoku board. The board is completed when each row, column, and 3x3 cube on the board has only one occurrence of numbers 1-9.

When the program asks "What cell would you like to edit ([row][column])?", the user can select a cell on the board by typing the cell's row and column characters and pressing Return.

Alternatively, if the user wishes to save the game and return to it at a later time, he/she may type 'qq' when prompted to enter a coordinate. This will save the current instance of the board in the Saved_Boards folder.

Once a value is entered into a cell, the game checks that value against the puzzle's solution. If there is a discrepancy between the entered value and the solution, the following message will appear:

```
"Hang on. The value you entered at [rowHead][colHead] isn't  
quite right."  
Try entering a different value."
```

The game ends when the current board's values exactly match their solution's values.

Known Issues

The path parsing in the loadSave algorithm is designed with Mac in mind. It's possibly that loading a save might fail on other Oses.

Occasionally, choosing to load a save from the main menu will not work, and the game will load a board as normal.