1 Documented Design: Sudoku Maker/Solver

1.1 High level Overview

The objective of this project is to create, automatically solve and grade (according to difficulty) randomly generated Sudoku puzzles. An easy to use graphical user interface is provided which allows a user to solve these puzzles, provide hints and maintain user scores.

A Sudoku puzzle is a logic-based, combinatorial number-placement puzzle. The objective is to fill a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 subgrids that compose the grid (also called ”boxes”, ”blocks”, ”regions”, or ”subsquares”) contains all of the digits from 1 to 9. The puzzle setter provides a partially completed grid, which for a well-posed puzzle has a unique solution. (reference: `https://en.wikipedia.org/wiki/Sudoku`)

The architecture consists of a server and one or more clients. In the server a Sudoku puzzle generator creates valid 9 by 9 Sudoku grids along with a unique key to identify each puzzle. This unique key is used to identify each puzzle and to eliminate duplicate grids. The grid and unique key are stored in a SQL database that is resident on the server. This SQL database is also used to record a user high scores table.

The application in the client provides the user interface for the Sudoku puzzle. This client application retrieves completed Sudoku grids from the server and then, using the games difficulty setting, it eliminates digits from the Sudoku grid to create a unique Sudoku puzzle of the appropriate level of difficulty.

Once the user has started to solve the puzzle, the partial solution is stored on persistent storage in the client machine so that the puzzle can be resumed after a break in play. The user can also request hints to solve the puzzle. Once a puzzle has been completed the user can submit their results to the server which will include: level of difficulty, time taken, guesses made and hints used.

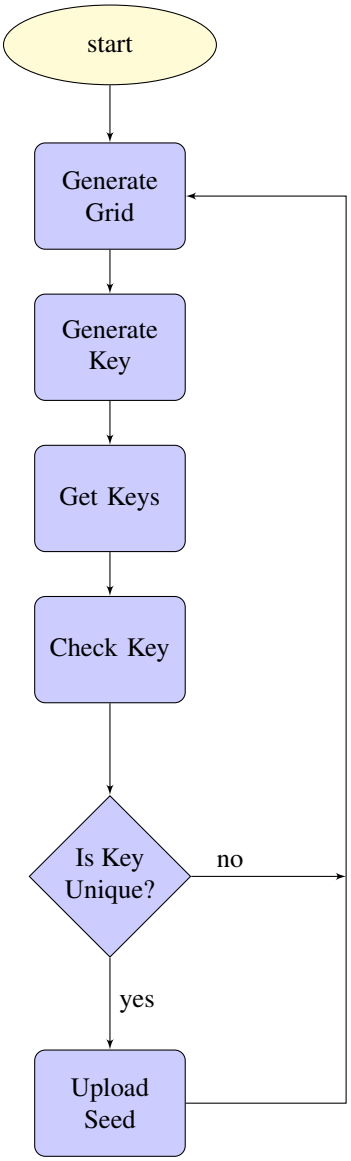
1.2 Server Side

1.2.1 Class: Node

This class (Node) is used to calculate a unique key for each Sudoku grid. The class uses a tree structure to hold all the trivial variations of a Sudodu grid including rotations, transformations, column and row switching. These trivial variations are created by simply swapping digits in the grid so can be regarded as the same puzzle from a Sudoku standpoint. A rotation is obtained by rotating all digits in the grid by a right angle (swapping x and y values); a transformation is obtained by swapping digit values (eg. All one digits swapped with all two digits); column and row swapping is achieved by swapping adjacent sets (sub squares) of 3 columns or rows.

When a new Sudoko grid has been created, a tree structure that contains all trivial variations of this new grid is computed, this is called the key. The leaf nodes of this key are compared with the previously computed keys to ensure that the new Sudoku grid is unique (except for the trivial variations). If any of the leaf nodes match the grid is discarded.

1.3 Key procedures



1.4 Generate Grid

This procedure will generate a complete Sudoku Grid. The procedure uses a back-tracking algorithm to create a valid grid. It starts at the first row and column and inserts a random digit (0 to 9). For each subsequent tile (digital location) a randomly shuffled list of digits (0 to 9) is used to select the next candidate for insertion. This list is maintained with each tile location. If the next candidate produces a valid grid the digit is inserted and the next tile is processed. Each digit from the list is tested, if no digit can be found that produces a valid grid the algorithm back-tracks to the previous tile and tries the next digit from the list for this tile until it successfully fills the entire grid.

1.5 Generate Key

This will generate a Key (all trivial variations of the grid) from a Sudoku grid. The Key will be generated by making a Tree using the Node class of all of the possible trivial transformations of the Sudoku seed.

1.6 Get Keys

This will return all of the Sudoku Keys from the SQL server. This is used to test for uniqueness of a new key.

1.7 Check Keys

Function: Upload Grid

This function uploads the grid and its variations from the tree to the SQL server. The variations are converted from the tree structure to a plain text format by in order traversal algorithm.

Function: High Score Manager

This function manages the high score table. When the high score table is updated the records are sorted by score and only the top five scores are retained.

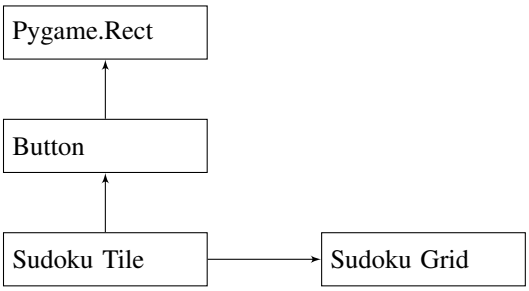
1.8 SQL server (Raspberry pi)

A Raspberry pi is used for the server in this project including hosting the SQL server. Consideration was given to internet hosting however since the compute load is relatively light this was not thought necessary.

2 Client Side

2.1 Classes

2.1.1 Inheritance diagram



2.1.2 Button

This class will directly inherit from Pygame.Rect. Pygame.Rect inherits from Pygame which is provided by the Python libraries. The Button class will:

- Take a text string and outline flag arguments that will create a text-based button with an optional outline easily.
- Take a function as an argument, allowing the button to transfer control to the function when the button is pressed.

The Button class is used for both normal game control and the digits within the Sudoku grid.

2.2 Sudoku tile

Sudoku Tile is a class that will directly inherit from Button and will exclusively be used in the Sudoku grid class. This will provide:

- A text argument (entered and displayed to the user) to assist solving the puzzle (this could be used to display possible values as small digits in each tile.)
- Functions to change the colour of the button/tile and to allow user input when selected.

2.3 Sudoku Grid

This class is a composition of Sudoku tiles, this class keeps track of:

- The Ids of each Sudoku tile that this class is managing.
- High score information.

This class also provides a method to enumerate the tiles in a row or column so that checks can be made to determine if certain moves are legal, and for the help function.

3 Key procedures

3.1 Main

This function provides the main loop for the program.

3.2 Main Menu

The user will be greeted by the main menu when launching the game. From this menu the user will be able to start a game, enter the options menu or exit out of the game. The user will also return to the main menu when they finish a Sudoku problem.

3.3 Draw Screen

This function draws the main screen, options menu and the Sudoku puzzle; the function is used to redraw the screen when necessary when the game is in progress.

3.4 Options

This function manages the options menu. The design goal for the options menu is to allow the user to change as many aspects of the UI (colours/settings) as possible. This includes the Difficulty settings of the game, the hints the user can obtain and the colours of all the graphical elements. In addition, this menu allows the user to enter complete Sudoku puzzles from other sources.

The Difficulty settings, within the options menu, controls various aspects of the game. These include the relative difficulty of solving the Sudoku puzzles presented to the user as calculated by the Sudoku generator; the amount of help the user is given, i.e. displaying all possible correct Values in tiles (hence eliminating the trivial incorrect Values); whether an incorrect move is accepted (this is important because an incorrect move in Sudoku makes the rest of the puzzle impossible to solve).

3.5 Game Menu

From this window:

- The user can select tiles in the 9x9 grid with left mouse button and input numbers with the on screen number selector, or through the Keyboard. By using the right mouse button the user can choose to input notes, this will allow the user to clearly see all the available Values for that tile.
- The user can ask for help, this will cause a message box to appear which will instruct the user how to continue with the Sudoku puzzle. This message box can be dismissed at any point if the user is satisfied with the help provided. Using this feature will disable recording of high scores, and a prompt will tell the user this.
- The user can ask the application to solve a Sudoku puzzle for them. The Game Menu will allow the user to input an external puzzle from another source (e.g. newspaper). The application will allow the user to solve this puzzle in the normal way or solve it automatically. This feature also disables recording of high scores.
- The user can choose to save and exit, if the user does this, the next time they start a puzzle they will be asked if they want to load their old puzzle or start a new one.

3.6 Import Seed

This will fetch the Completed puzzle from the SQL server and feed it into the Generate problem function.

3.7 Generate Problem

This will take a solved puzzle from the SQL server together with the Difficulty level and generate a Sudoku puzzle for the user to complete. It will do this by first performing a trivial transformation of the puzzle then randomly removing tiles and checking to determine whether the puzzle can still be solved. This uses the difficulty settings to determine which solving strategies must be used to solve the puzzle. The higher the difficulty setting the more advanced the set of strategies that will be required to solve the puzzle.

3.8 Submit Score to server

This will upload the users name, score, Difficulty and the Sudoku Key that was completed on to the server. The score will include how quickly the problem was completed (time) and how many mistakes were made.

4 Glossary

Sudoku grid

A Sudoku puzzle/problem is a 9x9 grid of Sudoku tiles.

Sudoku puzzle/problem

A Sudoku puzzle/problem is a Sudoku grid with only one solution for the user to solve.

Key/ID

A Key/ID is a unique identifier for a set of Sudoku puzzles/problems that have the same trivial transformations.

Completed puzzle/ Sudoku seed

A Completed puzzle/ Sudoku seed is a Sudoku puzzle/problem that has every single value filled in and obeying the Sudoku rules.

Tile (with reference to Sudoku grid)

A Sudoku Tile is an element of a Sudoku grid that can contain one non-zero single digit value.

Values (with reference to Tiles)

A Value is the Value that is stored within the tile

Dummy Values (with reference to Tiles)

A Dummy value is a technique that allows the user to list all remaining possible Values for a Tile

Difficulty (with reference to Sudoku grid)

Difficulty, in the scope of my project will define the level of Techniques that are required to complete the puzzle e.g Dummy Values

Techniques

A technique or strategy is a method of solving a Sudoku problem i.e.Dummy Values, BUG, X_Cycles, Unit_Forcing_Chains & Sword_Fish_Strategy

Transforming (a Sudoku grid)

Transforming or Transformations of Sudoku grids are the trivial ways one Sudoku puzzle can be transformed. i.e. a rotation of 90^circ

A unique Sudoku seed (with reference to the SQL server)

A unique Sudoku seed that cannot be trivially transformed into any of the already uploaded seeds.