Yatlongstan APCS
pd8
Yat Long Chan, Gabriel Thompson, Weichen Liu Spring 2022

Final Project Proposal
5UD0KU

Overview

We plan to make Sudoku in Processing. This project will include a board that we can interact with to play Sudoku. We will also implement a backtracking algorithm that will solve the puzzle.

Tools We Might Use

While nothing is final, and our vision so far is just a work in progress, we predict that we'll be using the following topics/tools:

- 2d array - this will be used to store the board.
- Recursive Backtracking - this will be used to make the puzzle solver.
- Stack - represent the call stack of the recursive backtracking, visually
- Processing - this will be used to make the visuals. It will also allow it to be more user-friendly.
- Processing User Input - so that the user can select different options, and enter in a board for the computer to solve
- Class Inheritance - store a parent class of both the normal sudoku board, and the hexadecimal sudoku

How It Will Work

To represent the board, we will use a "Board" class. This class will store two 2-dimensional arrays, one storing ints, and one storing booleans. The former will store the current board state. The latter will store which tiles are modifiable (because some of the sudoku tiles are unchangeable).

The Board class will also have methods. It will have two constructors, one of which will take in as arguments two ints, representing the width and height of the board. The other constructor will take in no arguments, and will automatically set the board to 9x9 (the default). The Board class will also have a generateTiles(int) method, which will populate the board with the specified number of tiles, in a configuration that is solvable. The Board class will have a getTile(int, int) accessor method, which will return the value of the tile at the row and column specified, and a setTile(int, int,

int) mutator method which will set the value of the tile at the specified row and column.

There will also be a “Game” class, which will utilize the methods in the “Board” class, and the Processing canvas, to play the game. This class will have a “Board” object attached to it. It will have a constructor method which instantiates the aforementioned “Board” object with the appropriate arguments. The “Game” class will also have a method to display the board as it is on the canvas, and will use Processing’s mouseClicked() (https://processing.org/reference/mouseClicked_.html) method to detect when the player clicks on different buttons, or selects a different tile. The “Game” class will use the keyPressed() function (https://processing.org/reference/keyPressed_.html) to take user input for writing numbers to the sudoku board. There will be a play() method with the gameplay described below.

Gameplay

1. You are presented with an option to either solve a sudoku board that the computer generates or create a sudoku board for the computer to solve.
2. If you chose to play a randomly generated sudoku board,
    a. Generate a board
    b. While the board isn’t solved,
        i. Allow the user to enter a value for any tile on the board that is modifiable
        ii. If the user presses the “QUIT” button, quit or something
    c. Add score to leaderboard?? (possibly)
3. If you chose to create a sudoku board that the computer solves,
    a. Create empty board
    b. Display board
    c. While “SOLVE” button isn’t pressed,
        i. Allow user to enter values for any tile on the board
    d. Check if user created board is a valid board
        i. If not, ask user to look over the board
        ii. Otherwise, run the solve algorithm. For each step in the solve algorithm, display the function call in the call stack
4. If player presses quit game,
    a. Quit the Processing sketch