

CS 101

STAGE 2 – PROJECT REPORT

Batch 611

TEAM MEMBERS

Ajeet Singh

Alekhya Audi

Archita Dungdung

Ayush Lakhotia

Hersh Manek

Harshit Agrawal

Ruhee D’Cunha

BRIEF DESCRIPTION OF VARIOUS MODULES/PROGRAMS

When the program runs, a window is displayed in which there are two buttons – Sudoku game and Sudoku solver.

Function used – menuwindow

Clicking on the Sudoku game takes him to a new window for the game, which is described below in Part 1. Clicking on the Sudoku solver takes him to a new window for the solver which is described in Part 2.

Class used – SimpleWindow s

Sudoku game (Part 1):

Puzzles have been obtained from the internet and put into files, according to the difficulty - easy, medium and hard.

Class - sudokugenerator

When the user chooses the level, one among the stored puzzles is randomly displayed in the grid for the user to play.

Function used - showtime

The timer begins as soon as the puzzle is displayed and the user can start playing. The user is allowed to input digits using buttons, which have been displayed at the bottom of the window. The user can change his input by clicking on the same box again and using the buttons.

Used – mouseclickcallback function

We have also provided the user with hints i.e he can get the digit of a particular box he wants to from the system. He can obtain a hint by clicking on the box he wants the clue for, and then on the hints button on the right side of the window. The clue is displayed in blue.

Function used - Hint

When the user finishes solving the Sudoku puzzle, he clicks on the “Done” button. The score is based on the time. The lesser time the user takes to solve the puzzle, the higher the score.

Function used – backtrack & is_legal

We invoke the Sudoku solver and it checks if the solved Sudoku is the correct solution.

Function used - correct

If it is the correct solution, a window is displayed which tells the user that he won and shows the time in which he solved the puzzle. If the user solves it within a given time, he achieves a high score, which is displayed in the window. The user can also quit a game anytime by clicking on the “Quit” button at the bottom right of the window.

Sudoku Solver (Part 2)

The user is given an empty 9X9 grid, in which he can input a Sudoku, which he needs a solution for. He can input using the buttons provided. When the user clicks on the “Solve” button after putting in the puzzle, the backtrack and the is_legal function is invoked and the solved Sudoku is displayed to the user.

STATUS OF COMPLETION

The game has been created. Hints have been provided. We haven't been able to check if the user is playing according to the rules after every digit he inputs. We finally check the solved puzzle at the end of the game. High scores have been calculated accordingly, taking into account the number of hints he takes and the time in which he solves.

The Sudoku solver has been created using backtrack. The puzzle given by the user is solved and displayed.

IDEAS FOR FUTURE WORK:

- There is scope for making a Sudoku generator, in which the digits in the Sudoku (depending on the level), will randomly be generated and placed accordingly in the grid.
- In the game, we could also check if the user is playing according to the rules after every digit he inputs.
- Also, we could highlight the conflicting entries (same digit in a row or a column or a 3X3 box) if there are any.
- A pause button could be done to pause the timer if the user wants to take a break in between the game.
- Saving a current game and loading a previously saved game could also be done.
- To calculate high score, we can also consider the number of times he puts in a wrong input.
- For high scores, we could allow the user to input his name and display the high scores in a separate window.
- We could also have better graphics, like changing the colour of the background.
- Input for digits can also be taken from the keyboard.

IMPORTANT ASSUMPTIONS

- The user cannot input a wrong Sudoku puzzle for the solver to solve.
- When the hints are given, they are meant to be guidelines and if the user chooses to use those values, he must input them himself. Each time the user clicks on “hint”, the hint count increases (irrespective of whether the user clicks on a box and then clicks hint, which actually gives the hint or just clicks on the hint button).
- If the user takes a hint for the same cell repeatedly, the hintcount increases.
- The user has to click on done when he finishes the game and quit, when he wishes to quit the game and exit at any point of time.

FUNCTIONS USED

Is_legal ()

Parameters are row, column and value. It goes to the particular cell and checks whether a digit corresponding to that value is present or not, in that row, column and corresponding 3X3 box.

Backtrack ()

The function sees to it that it changes only those values which are zeros(not inputed by the user) . It starts from the first cell and starts guessing values starting with 1. It uses the is_legal function to check whether the value is permitted and puts it in. It goes to the next row in the same column and repeats the same thing. When it reaches the last row, it goes to the first row next column, and keeps doing this till it reaches the 81st cell. When this happens, it returns 1. The backtrack function also updates the 2D array Solve[][] which is the solved Sudoku.

BRIEF DESCRIPTION OF INDIVIDUAL CONTRIBUTIONS

ACTUAL WORK DONE BY INDIVIDUAL PARTICIPANTS

Ruhee D’cunha

Intellectual contribution to the mouseclick function

Graphics – Creating windows, buttons, backgrounds, incorporating code into graphics

Ayush Lakhotia

Coding for sudoku solver, high scores, file input, correct function, documentation, graphics – grid

Archita Dungdung

Coding and graphics for timer, hints, position of boxes, windows

Ajeet Singh

Documentation, downloading, input of text files

Harshit Agrawal

Sudoku solver, checker, high scores, file handling, debugging

Hersh Manek

Coding for solver, checker, help with graphics, debugging errors, sequencing,

Alekhya Audi

Coding for solver, checker, documentation, file input, high scores, testing

CONSOLIDATED DIARY

Name	Discussion	Design	Programming	Testing	Documentation	Misc
Alekhya Audi	9hrs 30mins	-	14hrs 30mins	18 hrs	6hrs 30mins	7hrs 30mins
Archita Dungdung	6hrs 30mins	3hrs 30mins	7hrs 30mins	8hrs 30mins	5hrs 30mins	3hrs 30mins
Ajeet Singh	8hrs	-	-	3hrs 30mins	4hrs	-
Ayush Lakhota	12hrs	-	15hrs	18hrs	8hrs	4hrs
Harshit Agrawal	15hrs 55mins	-	15hrs	19hrs	2hrs	7hrs 30mins
Hersh Manek	11hrs 45mins	-	12hrs	24hrs	5hrs	6hrs
Ruhee D'cunha	5hrs	30mins	17hrs 30mins	13hrs	6hrs	4hrs 30mins

PEER REVIEW

Ruhee D'Cunha	111030005	10
Ayush Lakhotia	111030012	8.5
Harshit Agrawal	111030013	9.5
Archita Dungdung	111030019	8
Alekhya Audi	111030021	9.5
Hersh Manek	111030003	9.5
Ajeet Singh	111030027	3.5