Sudoku Solver

hk06665-or06360-rz06868

May 2021

1 Introduction

For this project we have made a Sudoku game with two main components:

1. The input and output
2. The Sudoku solver

We have used pygame for the input and output in order to make the game more user friendly (this is explained in detail below). And we used a combination of three functions to solve the Sudoku board which is represented as a nested list similar to the format of a graphs matrix representation.

2 Rules of Sudoku

Sudoku is a single player game the rules of which are fairly simple, the rules are listed below in order to make the Sudoku solver and the game more understandable. The game has three important rules:

1. Every square has to contain an integer from 1 to 9
2. Each 3×3 box can only contain each number from 1 to 9 once
3. Each vertical column and horizontal row can only contain each number from 1 to 9 once

The rules of the game are an integral part of both the solver and the game and will be seen in the explanation of both.
For the sake of this project a zero on the grid represents an empty square.
("How to Play Sudoku for Dummies – Every Rule Explained")

3 Pygame

We used Pygame in order to make the game more user friendly, and easier to follow. We used the following built-in functions:

3.1 pygame.event.get()

This function queues all the events made by the user ie. all the keys pressed by the user. Each key is associated with an integer number, for example esc is 27.

3.2 pygame.mouse.get_pos()

This functions returns the coordinates of the position of the mouse.

3.3 pygame.display.update()

This function updates the display of the entire pygame window after the changes made to it in a function.

3.4 display.blit()

Displays the said figure on the top of the last surface. This function is used to draw the rectangle grid for the Sudoku board display.

3.5 display.fill()

This function is used to fill the background colour, the function takes three parameters which are basically 3 numbers representing the colours red, blue and green. This function is used to fill in the background colour for the Sudoku game.

3.6 pygame.time.delay()

This function delays the function it is called in, the function takes one parameter which is the number of milliseconds they delay is for.

4 Sudoku Solver

To solve the Sudoku grid given the wide range of possibilities we used backtracking to solve the problem recursively using backtracking.

4.1 Backtracking

In order to fully understand the solution we must understand backtracking, what backtracking does effectively is it solves the problem recursively and simply eliminates solutions that do not fit a criteria. In the case of Sudoku the criteria is simply whether grid is set according to the game's rules.

4.2 Find empty

In order to find the next empty square to solve we simply use a while loop with the Boolean equation while

$$grid[i][j]! = 0$$

The loop changes the square in question until either an empty square is found or the grid is finished meaning that the grid has been solved.

4.3 Possible

This function is not only the basis of the entire solver it is also an important inspiration for the input. What the possible function does is it checks the row, column and square box of the square in order to check whether the number being entered is possible according to the rules of Sudoku.The function takes four parameters the grid, the row number, the column number and the number being entered.
The function has three for loops:

1. For loop to check the row:
   This for loop checks the entire row and breaks if the number is present in the row and returns false.

2. For loop to check the column:
   This for loop checks the entire column and breaks if the number is present in the row and returns false.

3. For loop to check square box:
   Since there are 9 square boxes, each with the dimensions 3X3, we used floor division to allot a square box to the position. Then we used a for loop with three iterations to traverse the square box. If the number is found the loop breaks and returns false.

4.4 Solve Sudoku

The Sudoku board is solved using backtracking the function looks for the next empty square. The function is called recursively, the position is filled in using a for loop with all the integers. The possible function is called and if the integer is possible to be placed in the position the position is filled with the said integer.

4.5 Solve Sudoku Visualiser

When pressed enter the function solve_vis(grid, i, j, display) is called which basically visualises ie. updates the display with each step.

5 Display

5.1 Draw board

We used this function to get a 9x9 grid for the Sudoku board.

5.2 Put numbers

To turn a 9x9 grid to a Sudoku board, we need to place numbers on it. We used this function to put in numbers from 1-9 at some places on the grid.

5.3 Instructions

For guiding the user on how to play the game or visualize the solver, we used this function to display the instructions. These instructions were: "Enter numbers 1 to 9", "Press 0 to erase your input", "Press enter to visualise" and "Press esc to get the solved Sudoku".

6 References

"How to Play Sudoku for Dummies – Every Rule Explained." Mastering Sudoku, Powered by Responsive Theme, masteringsudoku.com/sudoku-rules-beginners. Accessed 1 May 2021. "Learn Python by Building Five Games - Full Course." YouTube, uploaded by freeCodeCamp.org, 6 Nov. 2019, www.youtube.com/watch?v=XGf2GcyHPhc. "Python Programming Tutorials." Pythonprogramming, pythonprogramming, pythonprogramming.net/pygame-python-3-part-1-intro. Accessed 1 May 2021.