Federico Osti

Sudoku game

The project aims to code a Sudoku game on terminal using Python. The following points explain how it has been approached.

- Get a general idea on the game’s rules and user interface
- Sketch a flow chart of the program
- Create a class that describes each block in the flow chart
- Develop a finite state machine that defines the interactions of each block
- Handle user’s input exceptions

Flow Chart:

- **Update and display grid:**
  - Clear the terminal and display the matrix.
  - Display any exceptions or information.
- **Get user input:**
  - Prompt the user to input the row and column of the cell to edit.
  - Allow the user to input the desired digit.
- **Check user input:**
  - Validate the user input for character validity.
  - Verify if the selected cell is editable (initial digits in the grid cannot be edited).
- **Update matrix:**
  - Copy the user input into a 9x9 matrix.

- **Check if all cells are full:**
  - Determine if there is a digit in each cell of the matrix.
  - Indicate that the game is nearing completion.
- **Check solution:**
  - Examine the matrix to ensure it adheres to the game's rules.

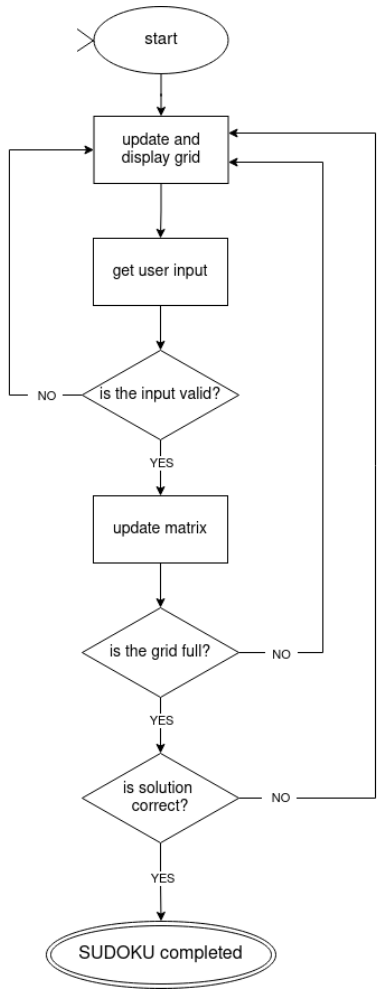

Image 1.1 flow chart

The code is designed to perform a singular task according to its current state, returning to the initial display block without employing nested functions. This architecture guarantees enhanced scalability for larger projects.

Object-oriented programming serves as a potent tool in the design process by enabling the initial specification of an object's properties at a lower level. Subsequently, from a higher-level perspective, it facilitates the design of interactions between individual blocks.

Git-hub

The project’s repository can be found here.

Sources

Chat-GPT has been used for displaying formatted text on the terminal and implementing the logic to verify the correctness of a Sudoku solution.