

1 Introduction

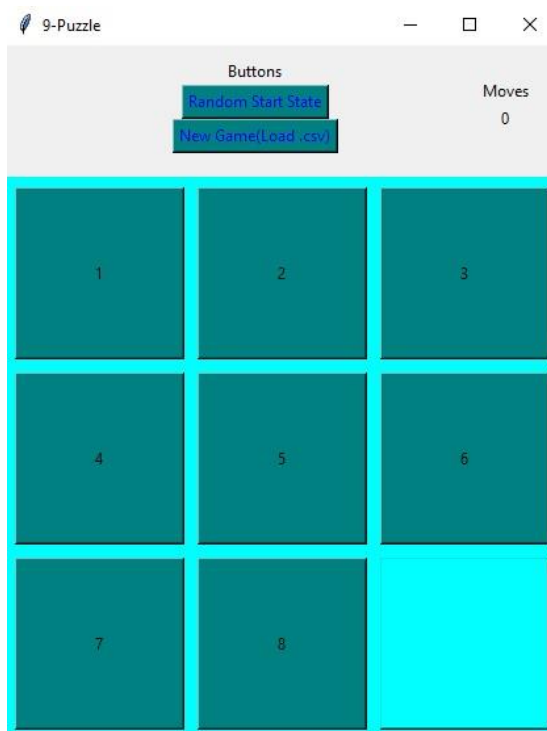
This is the documentation explaining the programming of a 9-puzzle game created using python. The purpose of this game is for a user to be able to solve a certain randomized number puzzle into the correct numerical form counting from 1 to 8 in the format of a 3x3 block. Furthermore, the game will count all the moves made by the user, and check whether the puzzle has been solved. Therefore, the only intelligence this program contains is to check if the puzzle has been solved by the user.

2 Literature study

For this program we used the python library Tkinter as it allowed for the easiest integration from the code of the 9-Puzzle to the GUI on screen. We looked at other methods to create a GUI for this program, such as PyGame, but Tkinter allowed us to have a basic GUI with minimal lines of code simplifying the process to display the 9-Puzzle. The puzzle makes use of various functions within Tkinter from the functions to section the GUI into different frames and allowed for the easy creation of usable buttons. Apart from Tkinter we also made use of the Random and csv libraries for specific functions.

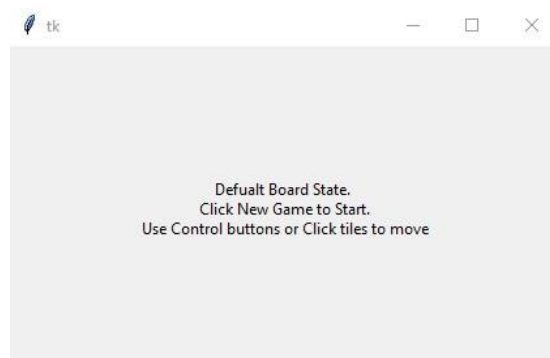
3 Program explanation

3.1 Default game board



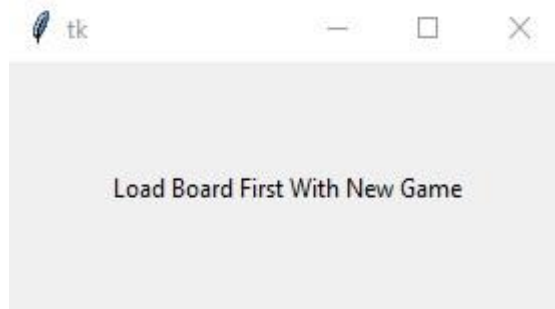
The default state of the game board is shown as the goal state. The GUI consists of eleven buttons which are grouped into specific frames, as well as three labels indicating the move counter and the buttons to load a new game. The tiles are made up of clickable buttons labelled 1 to 8 and includes the blank tile as well. At the top of the GUI are two main buttons which include: loading the new game from the csv file, which must be named "startstate.csv", and randomising a new start state. The moves made once the game has started will be tracked and displayed on the upper right of the GUI.

3.2 Starting state pop-up



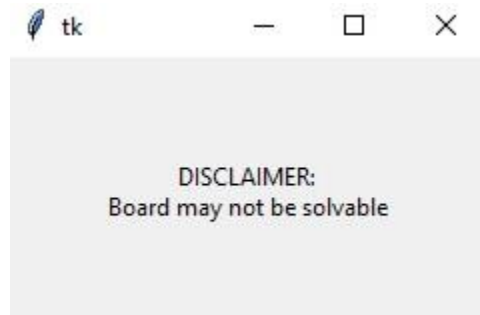
The game has a pop-up that appears with the default game board immediately when the exe file has been run. This particular pop-up informs the user that in order to play the game the "New Game(Load .csv)" button must be clicked for the game to work.

3.3 Default pop-up



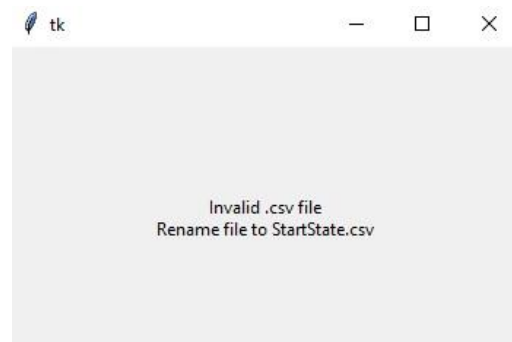
Once the game has been run, and the “New Game(Load .csv)” has not been clicked, any button clicked on the game board will cause this pop-up to appear. This pop-up reminds the user that in order for the game to work the “New Game(Load .csv)” must be clicked.

3.4 Randomised board pop-up

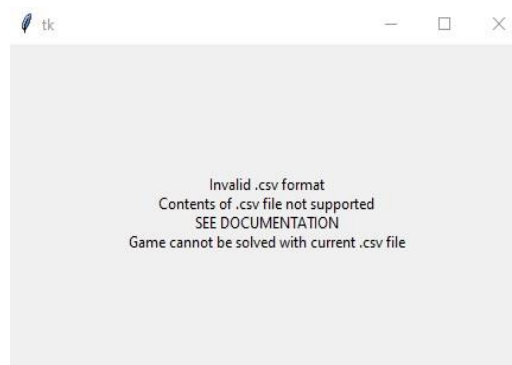


When the “Random Start State” button is clicked this pop-up will appear to inform the user that the current randomised game board is not necessarily solvable as there are some instances of a game board that are unsolvable.

3.5 Issue with csv file pop-ups

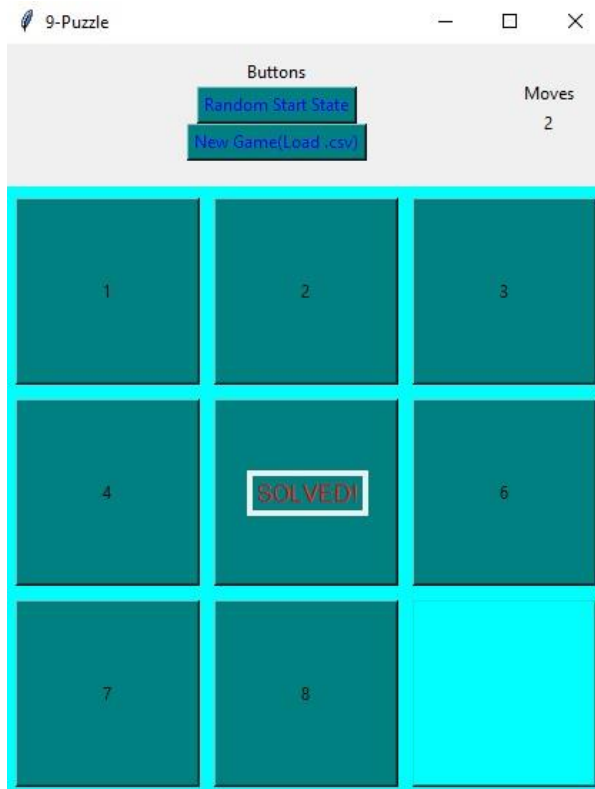


This pop-up appears when the program cannot find a csv file with the correct naming convention used in the code, i.e. “StartState.csv”, or if it cannot find a csv file at all.



This pop-up appears if the csv file is found but it cannot be converted properly into the program. The csv must contain only 9 elements ranging from 1 to 8 and the letter B. They can be in any order but must be delimited by a comma or not at all.

3.6 Solved game board



Once the puzzle has been solved by the user the game will display a label in the centre of the board indicating to the user that the puzzle has been solved. If the user wishes to play again they need just click one of the State buttons again.

4 Code Documentation (explanation of functions)

`puzzle()` – This is the constructor function of the program and calls on various other functions to set up the program. It creates the main outline of the GUI before calling additional functions to complete it.

`makeGUI()` – This function is responsible for creating the initial GUI for the program as well as the “behind-the-scenes” implementation of the puzzle in code format, i.e. a matrix. It creates and labels each of the tiles as well as sets up the buttons that will moves the tiles. It also creates the label to keep track of the moves as well as the buttons to load a start state for the game.

`csvToList()` – this function is responsible for reading in the contents of the csv file and implementing them into a board state to be used by the program. It also handles possible errors such as if the csv file is missing or incorrectly named and if it does not contain enough elements for a board state.

`loadStartStateGUI()` – this function is responsible to load the board with the state from the csv file. It calls the `csvToList()` function as well as the `updateBoard()` function to accomplish this.

`loadRandom()` – this function creates a random board state to be played. It also makes use of the `updateBoard()` function to reflect the board state on the GUI. This function is called on click of the “Random Start State” button.

`updateBoard()` – this function is responsible to update the GUI with whatever changes may occur to the “behind-the-scenes” puzzle matrix and reflect it on the GUI.

addMoves() – this function merely adds to the move counter and displays it on the GUI.

findBlank() – this function finds where the blank tile is on the board and returns its location as two numbers to signify where it is in the matrix. This function is used to determine whether a tile can or cannot be moved as tiles can only move to the blank tile.

checkSolved() – this function is responsible for identifying whether or not the puzzle has been solved and if it has displays the “SOLVED!” label on the GUI.

down() – this function moves the tile that is able to down on the board both in the matrix and the GUI. It is called on click of various tiles.

up() – this function moves the tile that is able to up on the board both in the matrix and the GUI. It is called on click of various tiles.

left() – this function moves the tile that is able to left on the board both in the matrix and the GUI. It is called on click of various tiles.

right() – this function moves the tile that is able to right on the board both in the matrix and the GUI. It is called on click of various tiles.

b1() – this function is the one used by the upper left tile and when that tile is clicked this function is called which will move the tile either down or right depending on which is legal.

b2() – this function is the one used by the upper middle tile and when that tile is clicked this function is called which will move the tile either down or left or right depending on which is legal.

b3() – this function is the one used by the upper right tile and when that tile is clicked this function is called which will move the tile either down or left depending on which is legal.

b4() – this function is the one used by the middle left tile and when that tile is clicked this function is called which will move the tile either up, down or right depending on which is legal.

b5() – this function is the one used by the centre tile and when that tile is clicked this function is called which will move the tile in whichever direction is a legal move.

b6() – this function is the one used by the middle right tile and when that tile is clicked this function is called which will move the tile either up, down or left depending on which is legal.

b7() – this function is the one used by the bottom left tile and when that tile is clicked this function is called which will move the tile either up or right depending on which is legal.

b8() – this function is the one used by the bottom middle tile and when that tile is clicked this function is called which will move the tile either up, right or left depending on which is legal.

b9() – this function is the one used by the bottom right tile and when that tile is clicked this function is called which will move the tile either up or left depending on which is legal.

newGame() – this functions is responsible for setting up a board state from the csv file using loadStartState() amd calls makeGUI() to reset the board completely, in case it has been solved. This function is called on click of the “New Game(Load .csv)” button.

5 Bibliography

Docs.python.org. 2020. *Csv — CSV File Reading And Writing — Python 3.8.3 Documentation*. [online] Available at: <<https://docs.python.org/3/library/csv.html>> [Accessed 6 June 2020].

Docs.python.org. 2020. *Graphical User Interfaces With Tk — Python 3.8.3 Documentation*. [online] Available at: <<https://docs.python.org/3/library/tk.html>> [Accessed 5 June 2020].

Docs.python.org. 2020. *Random — Generate Pseudo-Random Numbers — Python 3.8.3 Documentation*. [online] Available at: <<https://docs.python.org/3/library/random.html>> [Accessed 11 June 2020].