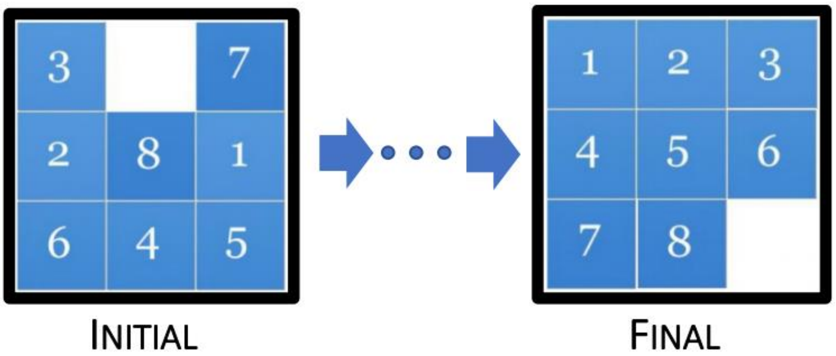
Design Doc

Overview

In this assignment, I design and develop an interactive sliding puzzle game where users can choose any dimensions up to 10x10, with minimum dimension 3x3.

The board has an empty space where an adjacent tile can be slid to. The objective of the game is to rearrange the tiles into a sequential order by their numbers (left to right, top to bottom) by repeatedly making sliding moves (left, right, up or down).

A 3x3 example shown below:



The program will end automatically if the puzzle is evaluated to be finished, it will also record the steps used to complete the game.

Data model

- **List:** the prototype of the puzzle is a one-dimensional list **(rawlst; newlst, manlst, flst are the variants of the rawlst for the purpose of print and evaluation)**.
- **Int:** the parameter **n** for the list **(n)**; the **index** of items in the list **(n1, n2)**; the **counter** for the steps **(count)** are stored as integers.
- **String:** the keywords for up/down/left/right movement are stored as strings **(up, down, left, right)**; some input prompts **(prompt2)** are also strings.

Program structure

(components)

1. Prompt the user to input the essential parameters that need to initialize the game. (puzzle scale, movement characters)
2. Generate the initial, ‘random’ puzzle.
3. Design four functions for the movement on each direction.
4. A printout function that can format and print the list as an n-by-n matrix.
5. A gaming part where the user can repeatedly input the instructions and get feedbacks.
6. A function that can evaluate whether the user has wined the game.
7. Restart/Exit the game by user’s instruction.

PROCESSING LOGIC (SPECIFIC)

- The **main function**: First call the functions prompt the user to input a number and 4 characters, stored them as **int (n)** and **str (up, down, left, right)** respectively; Generate the initial **list** using the **movement functions** and display it using the **printout function**; Call a **repetition function** that uses a while loop to prompt the user to repeatedly input instructions, match them with the reserved strings to execute the corresponding **movement functions**, use the **printout function** to display the result of each move, and use the **check-winning function** to decide whether to break the loop;
  Use another **while loop** which includes the main function to prompt to user to decide whether to exit or restart the game each time when the **main function** is finished.

- **Initial puzzle:** To make sure the puzzle is solvable, I first generate a sequential list containing n^2 -1 numbers and append ‘0’, which temporarily occupies the position for the blank space, to the last position of the list. Then I generate a random int in [1,4], each different int leads to the call of a movement function(up/down/left/right). I use a for loop to repeat this process n^4 times to have the initial, ‘randomized’ puzzle.

FUNCTIONAL SPECIFIC

1. chooseNum():
   # Ask the user to input a number n to certify the puzzle's scale
   # keep prompting the user to input again if the conditions (int, [3,10]) are not satisfied
   # declare n as a global variable

2. checkChar()
   # Ask the user to choose four characters to execute moving up, down, left, and right
   # keep prompting the user to input again if the input is not a character
   # Assign the inputs to up, down, left, right and declare them as global variables

3. createlist()
   # Generate a one-dimensional list by sequence and insert '0' at the last position
   # execute upwards/downwards/leftwards/rightwards function to the puzzle randomly by n^4 times
   # declare the list as a global variable and return it

4. upwards(), downwards(), leftwards(), rightwards():
   # Switch the position between '0' and the number up/down/left/right to it and return the outcome
   # return the initial list if there is nothing up/down/left/right to '0'
   # note that we accomplish the exchange by using the index difference between ‘0’ and the element up/down/left/right to it

# For each valid movement, update the counter to record the steps

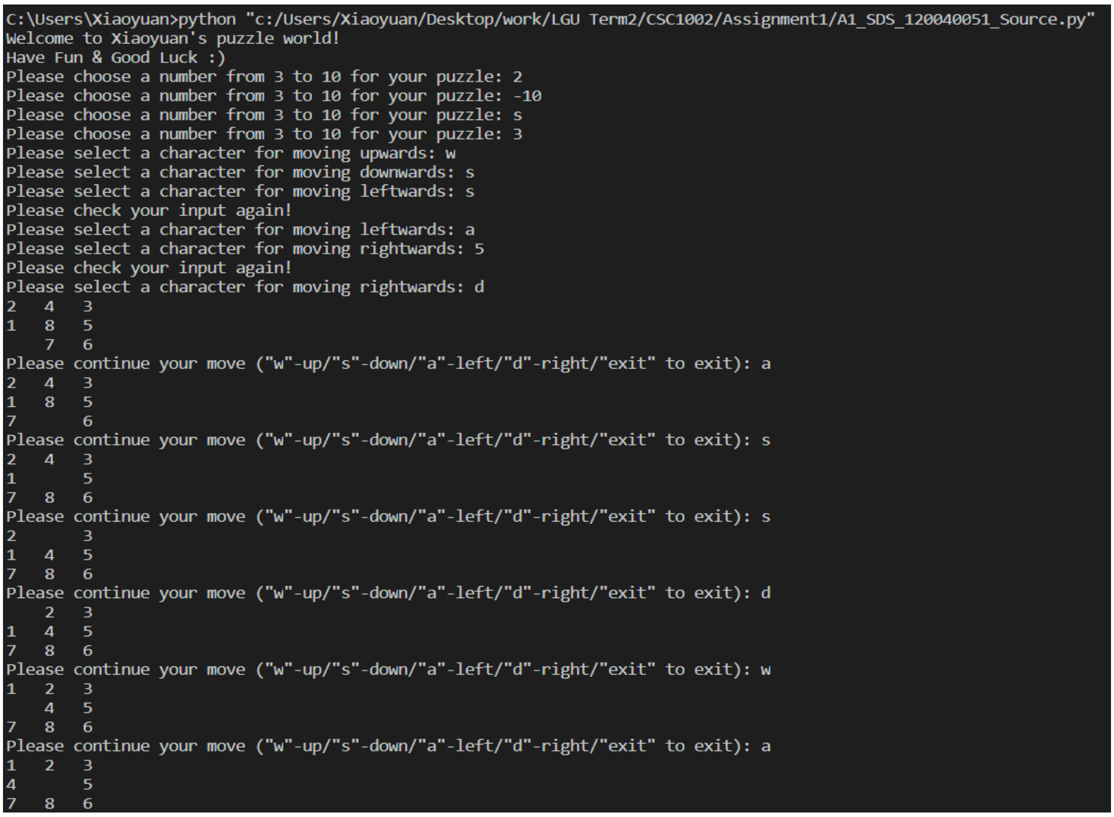
5. printout()
   # Replace '0' with a blank to form a new list
   # format the output by assigning 4 positions to each number
   # print the new list with n numbers a row

6. repetition()
   # Ask the user to execute the movement repeatedly until the final conditions are met
   # print the outcome for each movement
   # print a reminder if the input is not matched with the any chosen characters before and ask to input again
   # print the congratulations & steps used to complete if the result passes the evaluation function

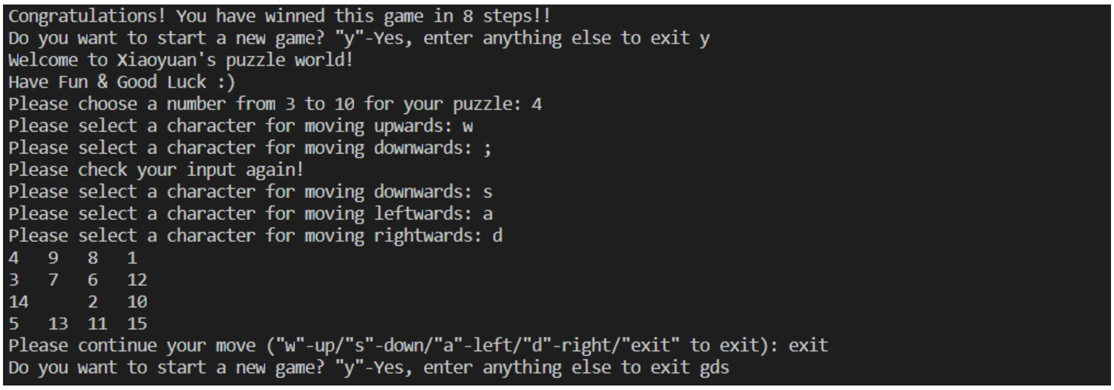
7. checkfinal()
   # check if the list is sorted by sequence

8. main()
   #call the above functions by logical sequence

SAMPLE OUTPUT



```
C:\Users\Xiaoyuan>python "c:/Users/Xiaoyuan/Desktop/work/LGU Term2/CSC1002/Assignment1/A1_SDS_120040051_Source.py"
Welcome to Xiaoyuan's puzzle world!
Have Fun & Good Luck :)
Please choose a number from 3 to 10 for your puzzle: 2
Please choose a number from 3 to 10 for your puzzle: -10
Please choose a number from 3 to 10 for your puzzle: s
Please choose a number from 3 to 10 for your puzzle: 3
Please select a character for moving upwards: w
Please select a character for moving downwards: s
Please select a character for moving leftwards: s
Please check your input again!
Please select a character for moving leftwards: a
Please select a character for moving rightwards: 5
Please check your input again!
Please select a character for moving rightwards: d
2   4   3
1   8   5
    7   6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): a
2   4   3
1   8   5
7       6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): s
2   4   3
1       5
7   8   6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): s
2       3
1   4   5
7   8   6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): d
    2   3
1   4   5
7   8   6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): w
1   2   3
    4   5
7   8   6
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): a
1   2   3
4       5
7   8   6
```



```
Congratulations! You have winned this game in 8 steps!!
Do you want to start a new game? "y"-Yes, enter anything else to exit y
Welcome to Xiaoyuan's puzzle world!
Have Fun & Good Luck :)
Please choose a number from 3 to 10 for your puzzle: 4
Please select a character for moving upwards: w
Please select a character for moving downwards: ;
Please check your input again!
Please select a character for moving downwards: s
Please select a character for moving leftwards: a
Please select a character for moving rightwards: d
4   9   8   1
3   7   6   12
14      2   10
5   13  11  15
Please continue your move ("w"-up/"s"-down/"a"-left/"d"-right/"exit" to exit): exit
Do you want to start a new game? "y"-Yes, enter anything else to exit gds
```