Design Document

Overview

This project offers a wide range of sliding puzzles in a variety of dimensions, from 3 to 10. Players can customize the keys to control the direction of movement.

Data Model

`number_list`: string	Store numbers in sliding puzzle
`n`: integer	The dimension of sliding puzzle
`row_and_col`: integer	Position of 0 or space
`directions`: list	Player-defined arrow keys
`direction`: string	Direction of movement

Program Structure

First, function `pos_int()` to get the dimension of player input. Function `init()` randomly generates a sliding puzzle in this dimension. During this, function `solvable()` to determine if the sliding puzzle has a solution. Ensure that the generated sliding puzzle has a solution, and output it as list `num_list`. Use the function `print_numbers()` to output a one-dimensional list as a two-dimensional array(sliding puzzle), and show `0` as a space. Function `option_limit_and_hint()` to get the position of a space in a number puzzle and determine and inform player the feasible direction of movement. Function `play()` moves the number by the

direction of the player's input. After success, the player can choose to continue the game or quit.

Processing Logic

Main processing logic

Use `init()` to initialization sliding puzzle. Use `while` loop to allow players to customize the direction of the four different button controls. In another `while` loop, first use `option_limit_and_hint()`, and then `play()`, finally `print_numbers()` to complete the move and display.

Initial Puzzle

Use `random.shuffle` to break up an ordered list(`[0,1,2…n**2-1]`). And determine if there is a solution by math. If there is no solution, repeat until there is a solution.

The sliding puzzle is sovable if and only if:

Dimension is odd, inverse order number is even;

Dimension is even, space in odd numbered rows (top row is the first row), inverse order number is odd;

Dimension is even, space in even numbered rows, inverse order number is even.

Ref: Wm. Woolsey Johnson, & Story, W. (1879). Notes on the "15" Puzzle. *American Journal of Mathematics, 2*(4), 397-404. doi:10.2307/2369492

Functional Spec

`pos_int()`: Use while loop and try/except to guarantee that the player enters an integer in `[3,7]`. Returns the final value entered by the player.

`init()`: Get n by `pos_int()`. Generate a list from `0` to `n**2-1` and randomly disrupt it with `random.shuffle` in to `num_list`. Determine whether there is a solution after random disruption by `solvable()`. `count` is used to record the number of player actions `n`, `num_list`, `count` are global variables.

`print_numbers()`: Output `num_list` as a numerical puzzle. Output `0` as a space. For aesthetics, fill number between 1 and 9 with a space.

`solvable(l)`: Finding inverse-order pairs using the dichotomy method. Determine if there is a solution by the above method. If there is a solution, return the Boolean value `True`. No solution return `False`. Parameters `l` is the list of judgments to be made.

`option_limit_and_hint()`: Use `num_list.index(0)` to find the location of spaces. Determine if the space is at a boundary and give limits and hints for player. Determine if the direction input by the player is feasible, if not, repeat the input until it is feasible. `direction` is a global variable.

`play()`: Moving numbers by `direction` input in `option_limit_and_hint()`. Actually swap the position of the `0` with the number in the opposite direction.

Sample Output

```
Enter the dimension of the number puzzle, which is an i
nteger in [3,10]:1
Error! It should be the integer in [3,10].
Enter the dimension of the number puzzle, which is an i
nteger in [3,10]:
```

```
Enter four different letters moving right, down, left a
nd right, separated by commas(,).dwdw,swsqwdqd,dwdxq
Error!
Enter four different letters moving right, down, left a
nd right, separated by commas(,).w,r,r,s
Error!
Enter four different letters moving right, down, left a
nd right, separated by commas(,).
```

```
Error! Please press the given key.
7  8  6
   5  4
1  2  3
Next:['Up', 'w']['Down', 's']['Left', 'a']
```

```
1  2  3
4  5  6
7  8
Congratulations! You have done 48 times to succeed!
Do you want to have another round?
Press 'y' to continue, press other keys to exit.y
Enter the dimension of the number puzzle, which is an integer in [
3,10]:4
7  10 5  11
14    6  2
15 1  13 4
12 9  8  3
Next:['Up', 'w']['Down', 's']['Left', 'a']['Right', 'd']
```