Design Documentation of Assignment 1

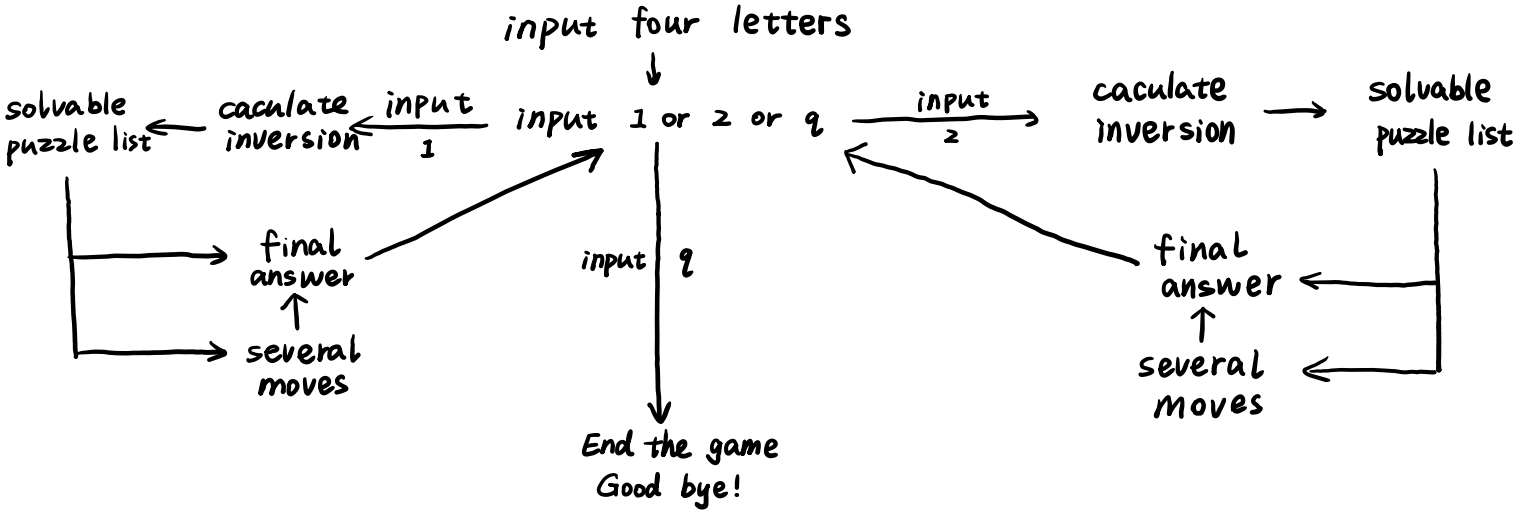
My thoughts and overall approach:

The user inputs 4 letters for left, right, up and down and choose 8-puzzle game or 15-puzzle game or quit the game. Then the program will randomly create a solvable puzzle. In the program, I use the list to represent the puzzle. There are nine kinds of ways to move the space and the program will instruct you to move. After several moves, if you reach the final form, you'll know how many moves you produce and have opportunity to quit the game or start new game.

Objects:

letters: the letters that user input
lst: a list which deletes the items in letters that are not letters
direction: a list contains four different letters
left, right, up, down: correspond to the four letters
time: the times of moving
puzzle_list: a list representing the puzzle
no_space_list: a list that doesn't contain space
choice: user inputs 1 or 2 or q to start the game or end the game

Structure:



Logic used to generate the puzzle:

How to make sure the puzzle is solvable? For 8-puzzle, it's solvable if the number of inversions is even. Inversion means a pair of tiles (x, y) if x appears before y but x > y. For example, the list [2, 1, 3, 4, 5, 6, 7, 8] has only one inversion (2, 1). For 15-puzzle, it's solvable if the space is on the first or third row and number of inversions is even, it's solvable if the space is on the second or fourth row and number of inversions is odd.

Use random.shuffle() to produce the randomized puzzle and use the method above to check whether it is solvable.

Functions:

end_the_game(): quit the game and return 'Good bye!'

move_up(i,N), move_down(i,N), move_left(i) and move_right(i): Four directions to move. 'i' means the position of the space in the list. 'N' means the overall number of row or column. These functions move the space through changing its position in the list. The times

of moving plus one after one move.

move(puzzle_list): Define the nine different kinds of move. When the space in different position, it has different ways to move

print_puzzle(puzzle_list): Print the puzzle in standard form, make sure to present the list in 3*3 or 4*4 form.

solvable(puzzle_list): Produce a solvable puzzle list

inversion(puzzle_list): Count inversions in the puzzle list. It will be used in function solvable(puzzle_list)

Output:

```
Welcome to this sliding puzzle program!
You need to choose either the 8 or 15-puzzle to play with!
You move the tiles with the keyboard using any 4 letters of your
own choice such as letters 'l', 'r', 'u', 'd' for left, right, up
and down moves respectively.

Enter the 4 letters for left, right, up and down: llll
Please enter four different letters!
Enter the 4 letters for left, right, up and down: lrud
Use letters l r u and d for left, right, up and down moves.
Enter 1 for 8-puuzle, 2 for 15-puzzle or q to end the game: 1
       2   5
   6   8   3
   7   1   4
Enter move (down-d,right-r): d
   6   2   5
       8   3
   7   1   4
Enter move (down-d,right-r,up-u): d
   6   2   5
   7   8   3
       1   4
Enter move (up-u,right-r): r
   6   2   5
   7   8   3
   1       4
Enter move (up-u,right-r,left-l): u
   6   2   5
   7       3
   1   8   4
Enter move (up-u,down-d,right-r,left-l): l
   6   2   5
       7   3
   1   8   4
Enter move (down-d,right-r,up-u): d
   6   2   5
   1   7   3
       8   4
Enter move (up-u,right-r): r
   6   2   5
   1   7   3
   8       4
Enter move (up-u,right-r,left-l): u
   6   2   5
   1       3
```

…….

…….

.....

```
   4   8   6
Enter move (up-u,down-d,right-r,left-l): r
   1   2   3
   5   7
   4   8   6
Enter move (down-d,up-u,left-l): u
   1   2
   5   7   3
   4   8   6
Enter move (down-dleft-,l): d
   1   2   3
   5   7
   4   8   6
Enter move (down-d,up-u,left-l): d
   1   2   3
   5   7   6
   4   8
Enter move (up-u,left-l): l
   1   2   3
   5   7   6
   4       8
Enter move (up-u,right-r,left-l): ul
Enter move (up-u,right-r,left-l): u
   1   2   3
   5       6
   4   7   8
Enter move (up-u,down-d,right-r,left-l): l
   1   2   3
       5   6
   4   7   8
Enter move (down-d,right-r,up-u): d
   1   2   3
   4   5   6
       7   8
Enter move (up-u,right-r): r
   1   2   3
   4   5   6
   7       8
Enter move (up-u,right-r,left-l): r
   1   2   3
   4   5   6
   7   8
Congratulations! You solve the puzzle in 86 moves.
Enter 1 for 8-puuzle, 2 for 15-puzzle or q to end the game: q
Good bye!
```