CSC1002
Assignment 1 Design Document

Chen Ang
118010009

March 2019

1 Programm Structure

```
Initialize $puzzle$ randomly
$step \leftarrow 0$
repeat
    Display $puzzle$
    Input: sliding direction $D$
    Update $puzzle$ by $D$
    $step \leftarrow step + 1$
until
  $puzzle$ is solved
Output: $puzzle, step$
```

2 Python Objects

2.1 int

Used to locate tiles and count steps

2.2 list

Used to store and edit the puzzle.

2.3 dictionary

Used to replace an if-elif-else statement.

2.4 string

Output messages and user input are strings.

3 Functions

3.1 swap(list, index_1, index_2)

Swaps the value of two list items.
list: the target list
index_1, index_2: indices of target items

3.2 solvable(puzzle)

Returns True if puzzle solvable, False otherwise.
puzzle: an 8-puzzle as a list

3.3 generate_pzl()

Returns a *solvable* 8-puzzle as a list.

3.4 print_pzl(puzzle)

Prints out puzzle in 3×3 form.
puzzle: an 8-puzzle as a list

3.5 solved(puzzle)

Returns True if puzzle is solved, False otherwise.
puzzle: an 8-puzzle as a list

3.6 possible_dir(puzzle)

Returns all possible sliding directions(str) given puzzle.
puzzle: an 8-puzzle as a list

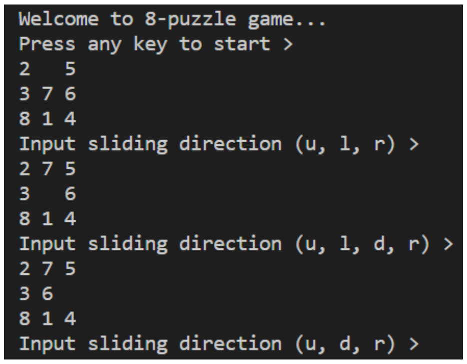
3.7 slide(puzzle, direction)

Slide puzzle in given direction. Returns a new puzzle as a list.
puzzle: an 8-puzzle as a list
direction: direction in which the tile slides
"u": up, "d": down, "l": left, "r": right

3.8 new_game()

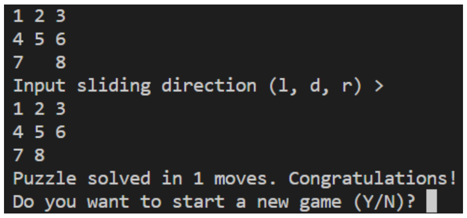
Starts a new game.

4 Sample Output



```
Welcome to 8-puzzle game...
Press any key to start >
2   5
3 7 6
8 1 4
Input sliding direction (u, l, r) >
2 7 5
3   6
8 1 4
Input sliding direction (u, l, d, r) >
2 7 5
3 6
8 1 4
Input sliding direction (u, d, r) >
```

Figure 1: Gaming



```
1 2 3
4 5 6
7   8
Input sliding direction (l, d, r) >
1 2 3
4 5 6
7 8
Puzzle solved in 1 moves. Congratulations!
Do you want to start a new game (Y/N)?
```

Figure 2: End of the Game