Design Doc

1.Overview

This is a program about “Sliding Puzzle”. My program will allow users to enter the four keys they want to control the number to move into the space. Then let the users choose the length of the square that they want. After being messed up, the users can play the game. And print out how many steps have been made after the game is over.

2.Data Model

I used many parameters in the program to help the game progress.

Parameter:

1. number_list_of_square_in_dimension: This parameter is the real-time order of numbers in the list. I use the list to store the information of the square.
2. right_order_list: This parameter is the correct order of numbers in the list.
3. moving_times: The parameter "moving_times" will count the numbers of moving step.
4. move_to_which_direction: I use the number to represent the direction. So, the parameter "move_to_which_direction" shows which direction the number should move to.
5. mess_times: The parameter “mess_times” shows how many times random move have been done. It’s used to mess up the square.
6. goingup/goingdown/goingleft/goingleft: I use these parameters to store the keys that represent how the number move. Users will input this information at the beginning of the game.

3.Program Structure

Structure:

1. Initialize the game and square.

·Run_Game(): This function is to initialize the list which store the information of the square. This function also play a role as main function. Most operation are happen in this function.

·print_square(playnumber, number_list_of_square_in_dimension): This function will print the square.

·move_control(playnumber, number_list_of_square_in_dimension, move_to_which_direction, moving_times): This function is to move the order of

numbers in the list, then the square will be changed.

2. Mess up the square.

·mess_the_order(playnumber, number_list_of_square_in_dimension, move_to_which_direction, moving_times): This function will help to mess up the order of the square. In the same time, it use move_control() function to mess up, so the square is always solvable.

3. Move and judge the ending.

·move_hint(playnumber, number_list_of_square_in_dimension): This function will print out which direction the number can move.

·The Run_Game(): will also judge whether the game is end by compare real-time list with right list.

4. Ask for another game.

A “While True” loop is used to ask the player to play it again or end the project.

3.Processing Logic

Main processing: the function Run_Game() will play a role as summary. It will include other function, and process the whole move and judge part.

Initial Puzzle: All square in my project is initialize with order. But I use the function mess_up() to mess the ordered square up. In the same time, it use move_control() function to mess up, so the square is always solvable.

4.Functional

###Run_Game(): See in Structure 1.

###print_square(playnumber, number_list_of_square_in_dimension): This function will print the square.

playnumber: Show how large the square is.

number_list_of_square_in_dimension: Show how the real-time list is.

###move_control(playnumber, number_list_of_square_in_dimension, move_to_which_direction, moving_times): This function is to move the order of numbers in the list, then the square will be changed.

playnumber: Show how large the square is.

number_list_of_square_in_dimension: Show how the real-time list is.

move_to_which_direction: Show which direction the number should move.

moving_times: How many steps have been done.

This function will return “number_list_of_square_in_dimension” and “moving_times”

###mess_the_order(playnumber, number_list_of_square_in_dimension, move_to_which_direction, moving_times):This function will help to mess up the order of the square.

playnumber: Show how large the square is.

number_list_of_square_in_dimension: Show how the real-time list is.

move_to_which_direction: Show which direction the number should move.

moving_times: How many steps have been done.

This function will return “number_list_of_square_in_dimension”

###move_hint(playnumber, number_list_of_square_in_dimension): This function will print out which direction the number can move.

playnumber: Show how large the square is.

number_list_of_square_in_dimension: Show how the real-time list is.

5.Sample Output

```
Welcome to Sliding Puzzle!
In this game, you can choose 3*3, 4*4 or enven more demension puzzle.
move your number and make 3*3 puzzle to be:
1       2       3
4       5       6
7       8
or make more demension puzzle to be in order.
enter the key for up:w
enter the key for down:s
enter the key for left:a
enter the key for right:d
Do you want to start a new game?(enter yes/no)yes
which do you want to play(enter 3 for 3*3、enter 4 for 4*4、or more):3
4               1
5       6       3
7       8       2
Enter w for up
Enter a for left
Enter d for right
Enter the direction:a
4       1
5       6       3
7       8       2
Enter w for up
Enter d for right
Enter the direction:w
4       1       3
5       6
7       8       2
Enter w for up
```

Enter s for down
Enter d for right
Enter the direction:d
4 1 3
5 6
7 8 2
Enter w for up
Enter s for down
Enter a for left
Enter d for right
Enter the direction:s
4 3
5 1 6
7 8 2
Enter w for up
Enter a for left
Enter d for right
Enter the direction:w
...
...
...
...
1 2 3
4 5
7 8 6
Enter w for up
Enter s for down
Enter d for right
Enter the direction:w
1 2 3
4 5 6
7 8
Congratuations, you use 114514 steps to finish it
Do you want to start a new game?(enter yes/no)yes
which do you want to play(enter 3 for 3*3、enter 4 for 4*4、or more):4
...
...
...
...
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15
Congratuations, you use 114514 steps to finish it

Do you want to start a new game?(enter yes/no)no
Thank you for playing!