**1.Design.**

**a.Thought:**

(1)Print the matrix randomly.So I import random and use the ‘random.sample’ to get a random list, with 1 empty space

and 8 or 15 integers. Then I print the elements in the list with 3 one line. So i print the question and offer it to the player.

(2)Check if the answer has answers. If not, we don't print it and restart(1)

(3)Check if the input is right.(4 different elements) First, get elements apart. And set up a list and we use 'for' loop to get them together. At the same time, we get some elements apart such as'df' into 'd' 'f'. finally, we use len()to check if there are 4 elements in the list.

(4)According to the different position of the empty space, we remind the player of different ways(left,right,up and down) We can make a 'while' loop to check if the player gives the right input. if not, it restarts and tell the player to input the way again. Don't forget that we use thelist to represent the matrix. so 'change' means we exchange two elements.

(5)We use a loop to After the change is made. if the list do not turn into[1,2,3,4,5,6,7,8,' '] or [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15], it means the problem are not solved and the loop continues. so we give the player the matrix changed. Else, the player has finished the game. He can start a new game or stop the game.

**b.Python objects**

Global variables:

1. matrix. A list to store the matrix, both 8-puzzle and 15-puzzle.
2. n1. A list to store the letter for moving.
3. Choice: a string to store the size of the map. ‘1’ for 8-puzzle. ’2’ for 15-puzzle. ‘q’ to stop the game.

b.Local variables:

Print\_matrix: s to calculate the number of numbers have been printed one line. Each time one element is printed, the value of s will be 1 bigger than before. If it turns into 3(or 4), then the value of s returns to 0. and we use print() to start the second line.

Position: s to represent the position of the empty space in the list.

Check0:

matrix2 is a copy of matrix. So we can delete the empty space without influencing matrix. S was used to calculate the number of ‘inverse numbers’ in the list.(if i >j and a[i]<a[j], that represents one inverse number.

Check2:

Choice: means the choice made by the player, it can be ‘1’, ’2’ or ‘q’

b is the list to contain the elements in n1.

Change\_fun(): change is used to contain the right ways to change the matrix.

Main(): b is used to check if the input is right. If yes, b=0. if no, b=1. change\_matrix means the possible changing ways. After that, it is used to calculate the times of the movement.

Posi: position of the empty space.

a: the changing way selected by the player

**c.structure:**

The main structure is as follows:

Start main() --> print instruction --> input for left, right,up,down -->check1() and check2()(if false it will go back to input) --> input for choice(1,2 or quit) --> if choice is not quit --> get a matrix(3\*3 or 4\*4) -->check0()(if false, we will get another matrix) --> print\_matrix(matrix) --> position(matrix) --> show the directions changing that can be done --> input the changing direction(if input is wrong, repeat it) --> changing the order in matrix --> if succeed, go back to input for choice. If not, go back to show the directions that can be done.

If quit, the game will be over.

**.logic of generating the matrix**

Use random.sample(matrix) to generate the matrix. Then, check whether it is solvable. The method is as follows:

1. Definition

For a list of numbers, such as n=[1,2,3,4,8,5,...]. if n[i-1]>n[j-1] while i<j, then the total number of the (i,j)we call this inversion.

1. Theorem
2. If it’s a 8-puzzle, then the question will have answer if the number of inversions is an even number.
3. If it’s a 15-puzzle, then the question will have answer in two situations.
4. the number of the inversions is an odd number and the empty space is in 1 or 3 th row.
5. The number of the inversions is an even number and the empty space is in 2 or 4 th row.

**4.Function:**

The main procedure consists of 7 functions:

1.print\_matrix(). It was used to print the matrix. it can turn the list into a matrix. Notice that it can deal with both 3\*3 and 4\*4.

2. 3 check function. if yes, it will return True

check0(). it is used to check if the question has answers.

chesk1(). it is used to check if the player inputs 4 elements.

check2(). it is used to chesk if the player inputs some same elements.

3.change\_fun(). it can return that the kinds of movement that can be done, according to the position of the empty space.

4.position().it is used to find the position of the empty space.

5.main(). it is used to design and start the game. Use a loop to keep the game process.