

Solving constraint satisfaction problems using RAT

16 March 2022



The Team

Sarthak S Kumar -PES2UG21CS432

Varun C -PES2UG21CS593

Vishal M Godi -PES2UG21CS607

Section: K

Batch: 2

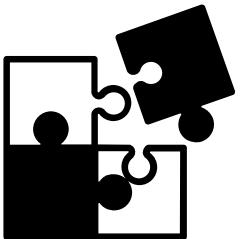


Constraint satisfaction problems?





It is the process of finding a solution to a set of constraints, that impose conditions that the variables must satisfy. A solution is therefore a set of values for the variables that satisfies all constraints.



Constraint satisfaction problems on finite domains are typically solved using a form of search. The most used techniques are variants of backtracking using **recursion**, constraint propagation, and local search.



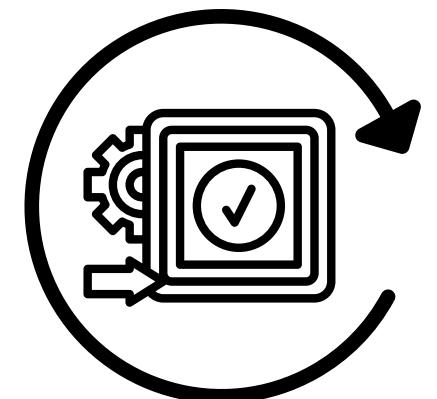
Recursive algorithms





A recursive function is one that calls itself with simpler values.

In recursion, we try to solve a problem by solving a smaller instance of the same problem, unless the problem is so small that we can just solve it directly.



Backtracking is a recursive tool for solving constraint satisfaction problems



Backtracking



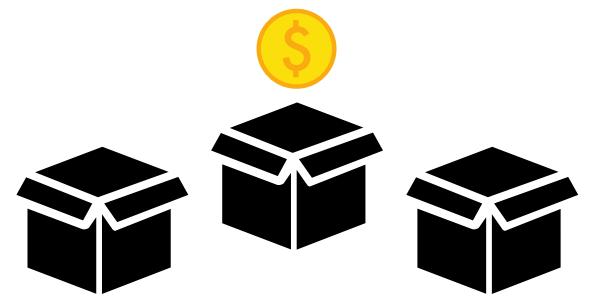
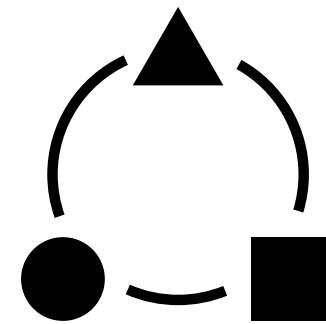


It's an **algorithmic-technique** for solving problems recursively by trying to build a solution incrementally, one at a time removing those solutions that fail to satisfy the constraints of the problem at any point of time

In short words, it's searching **every possible combination** in order to solve a computational problem.

In **recursion**, the function calls itself until it reaches a base case.

In **backtracking**, we use recursion to explore all the possibilities until we get the best result for the problem.



Implementation

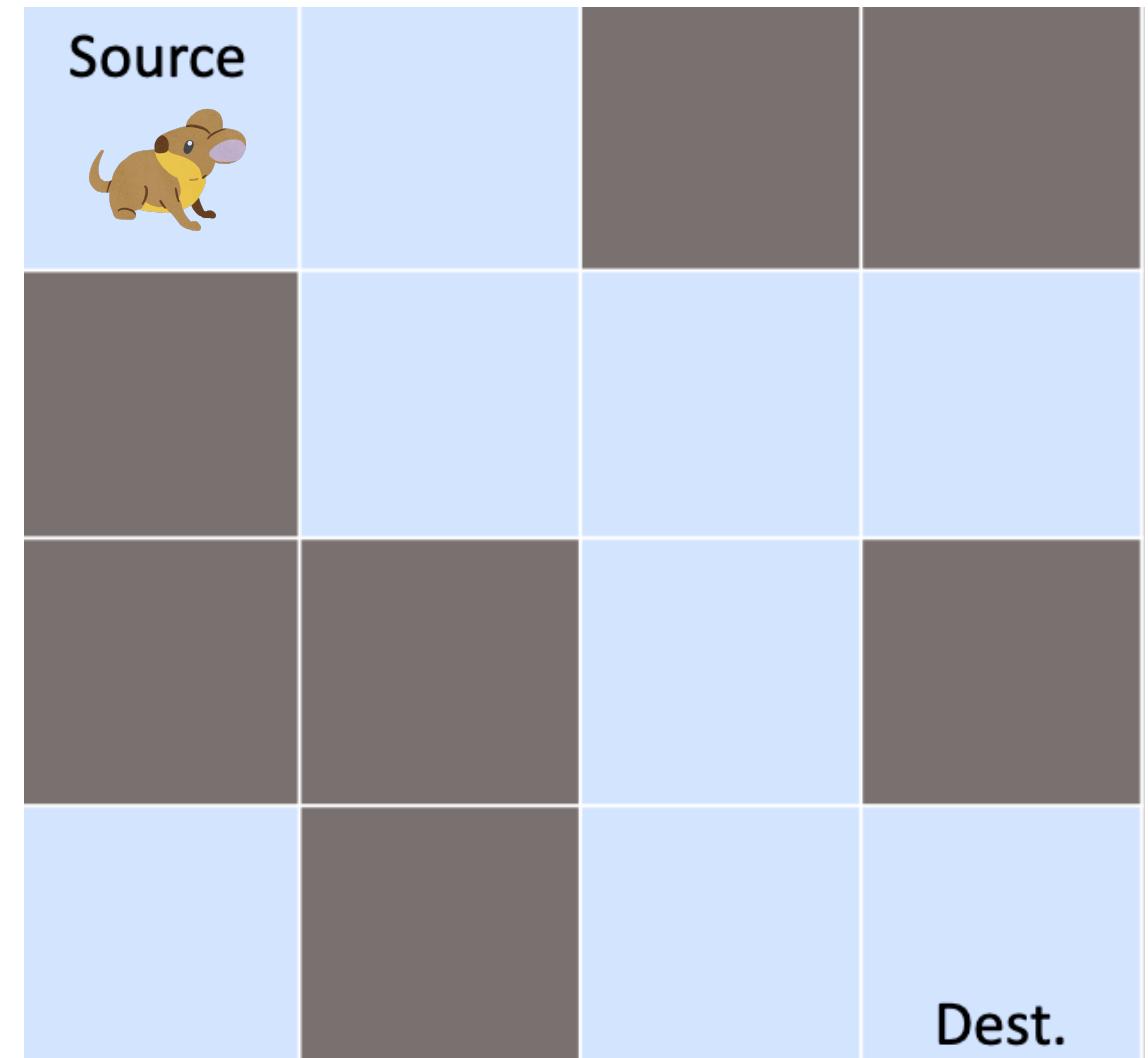


Rat in a maze

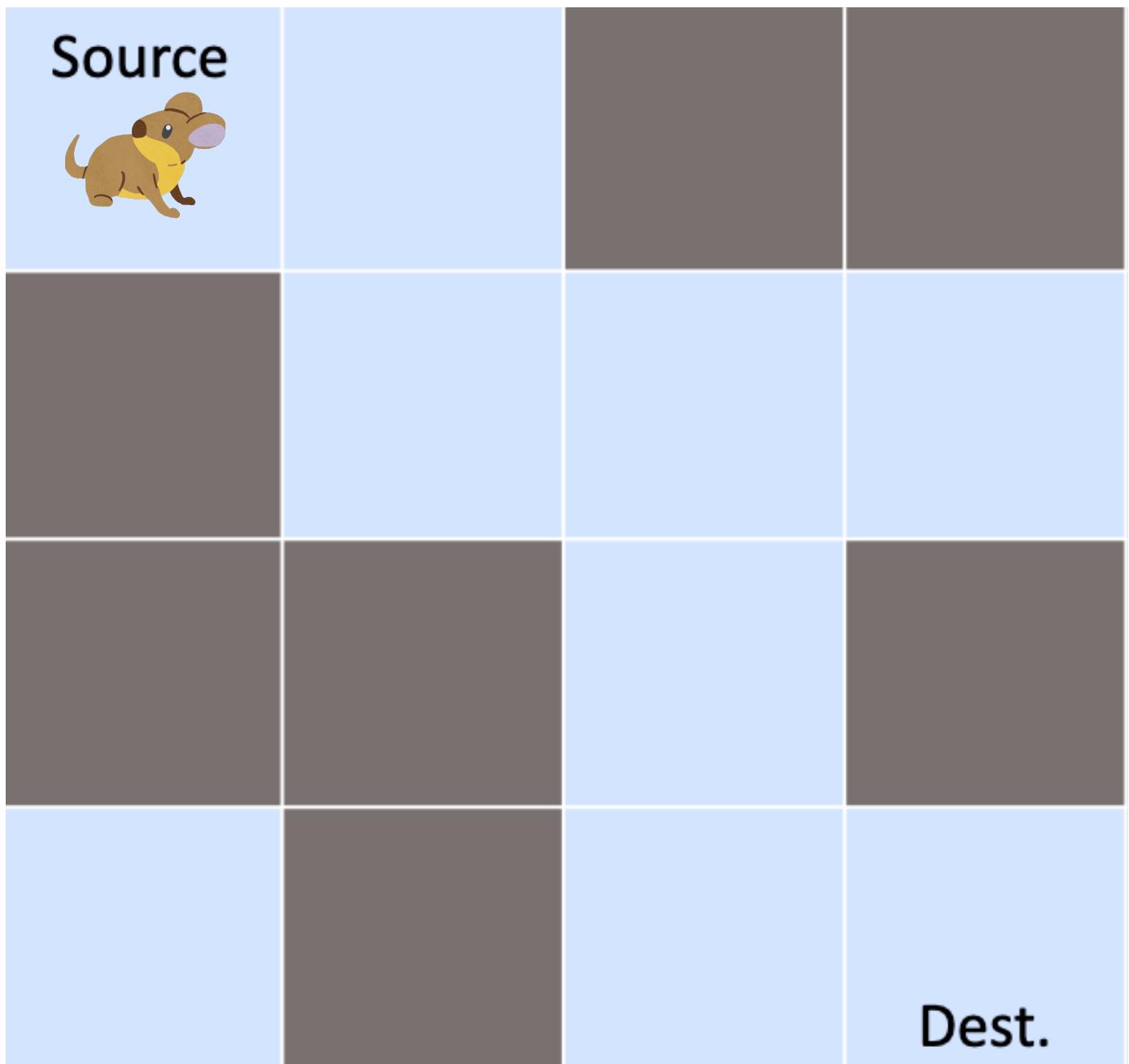


Constraints

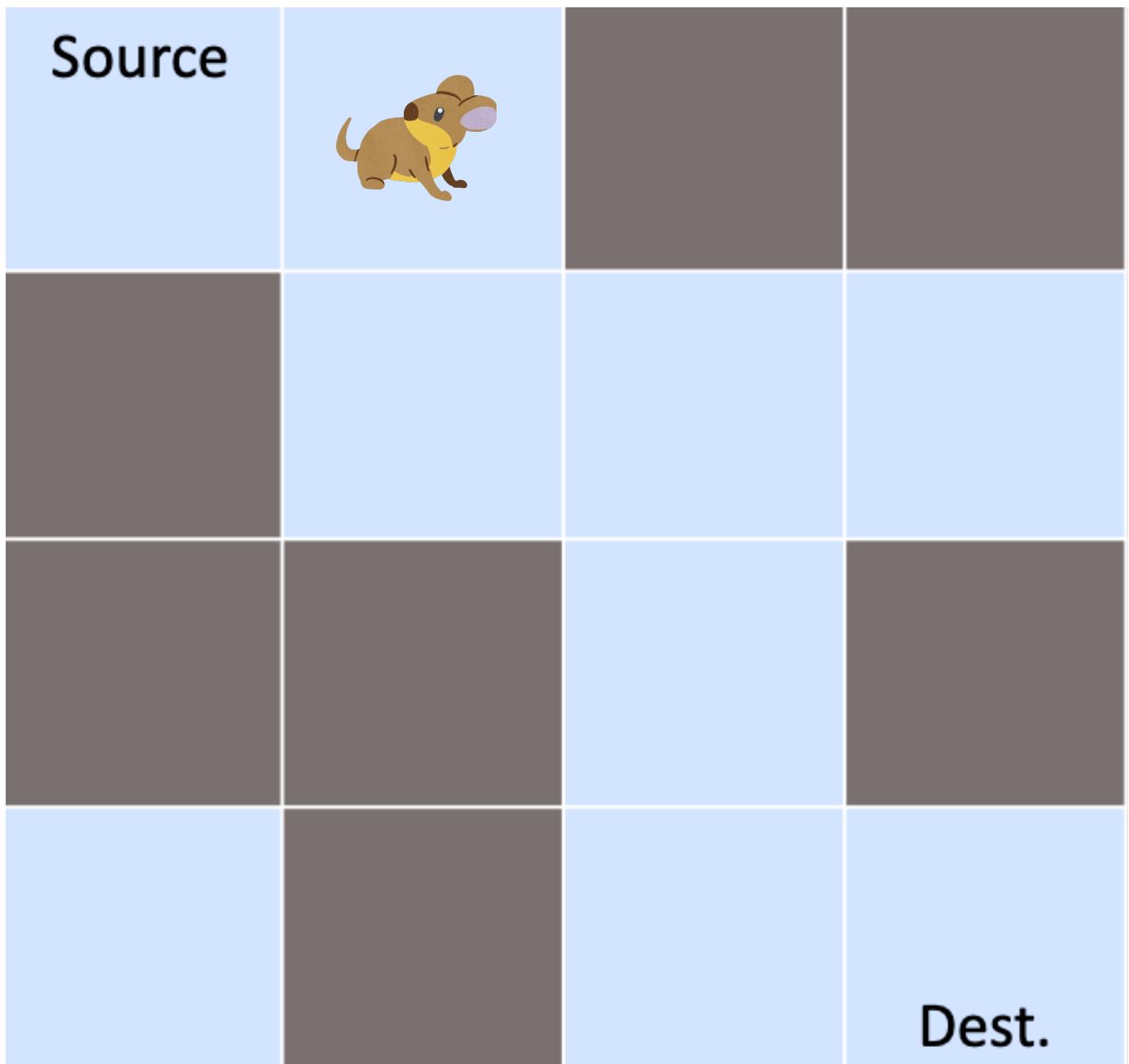
- The starting position is the tile on the first column and first row and the finishing position is at the last column on the last row.
- The rat can only travel on the valid tiles which are represented in white and cannot travel on the obstacle tiles represented in black
- The goal is to make the rat travel from the starting position to the ending position



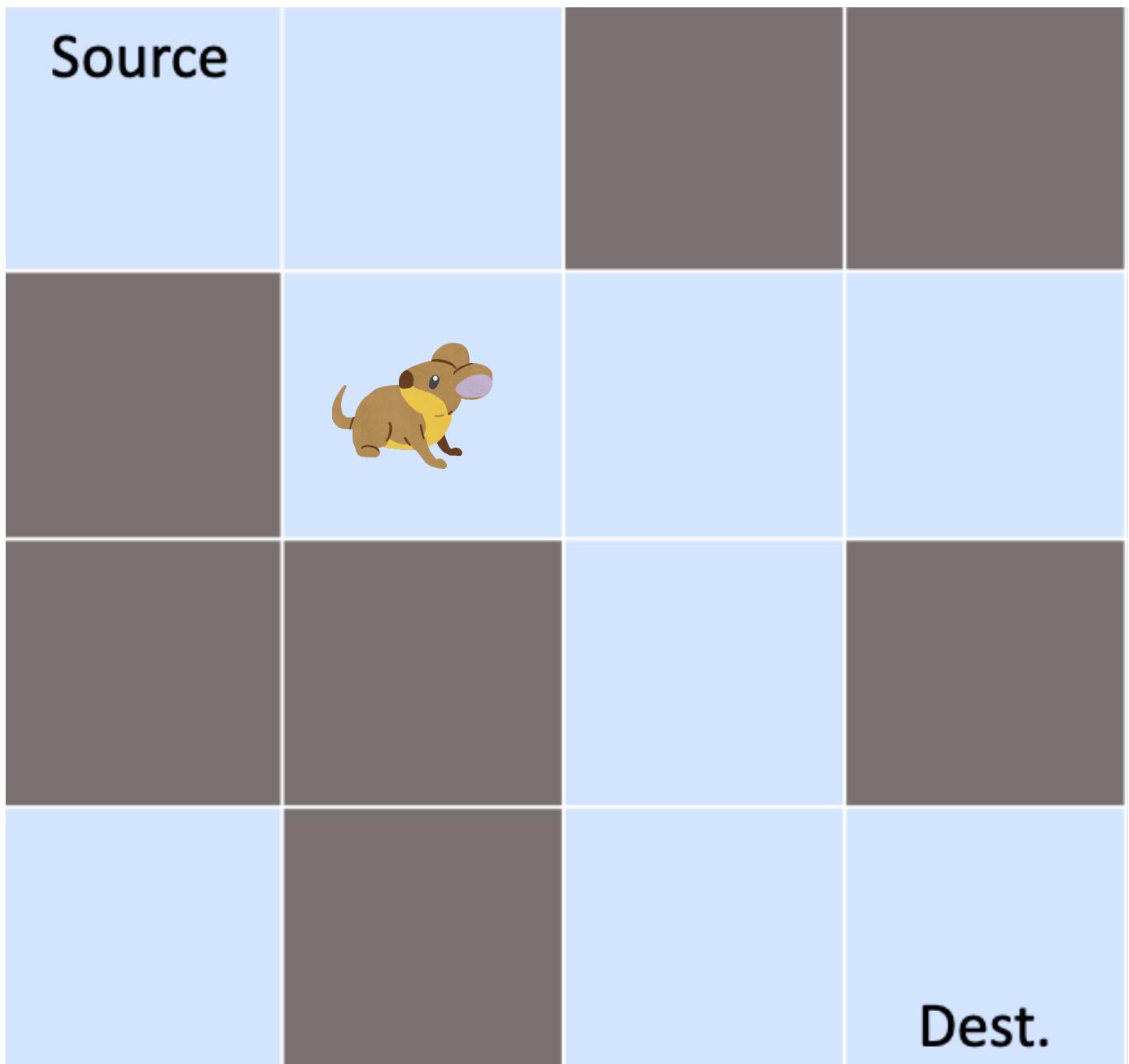
Rat in a maze



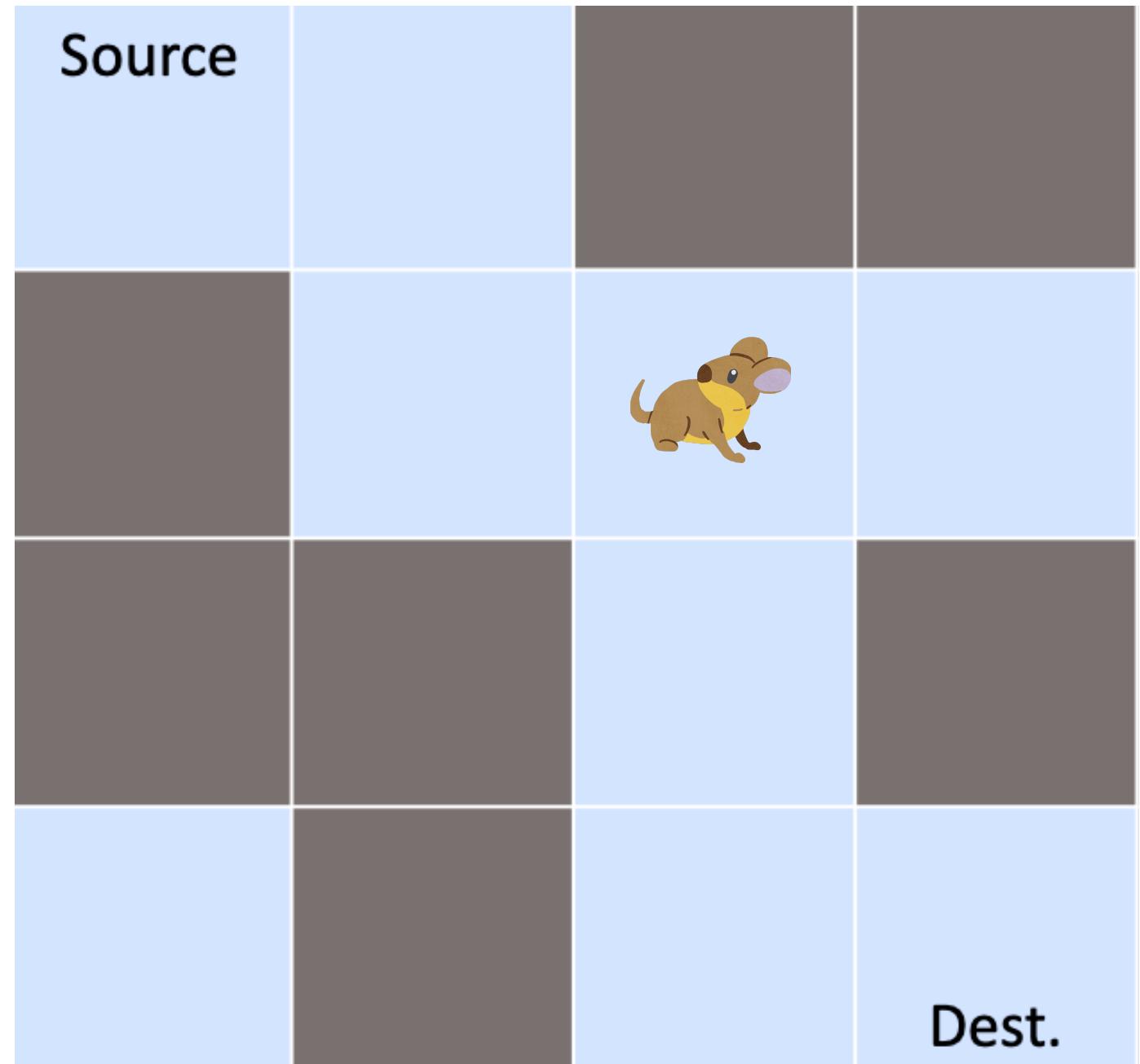
Rat in a maze



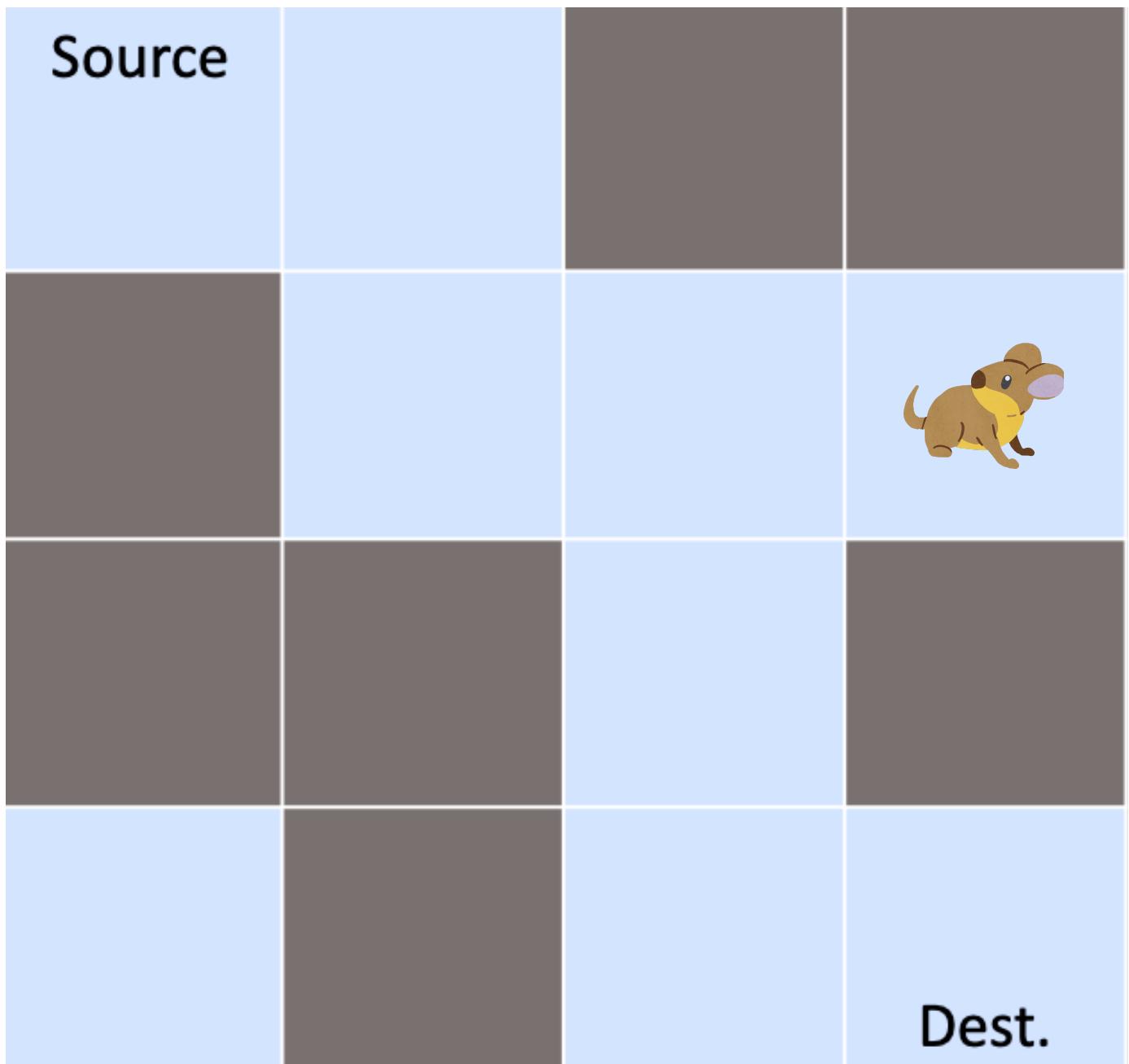
Rat in a maze



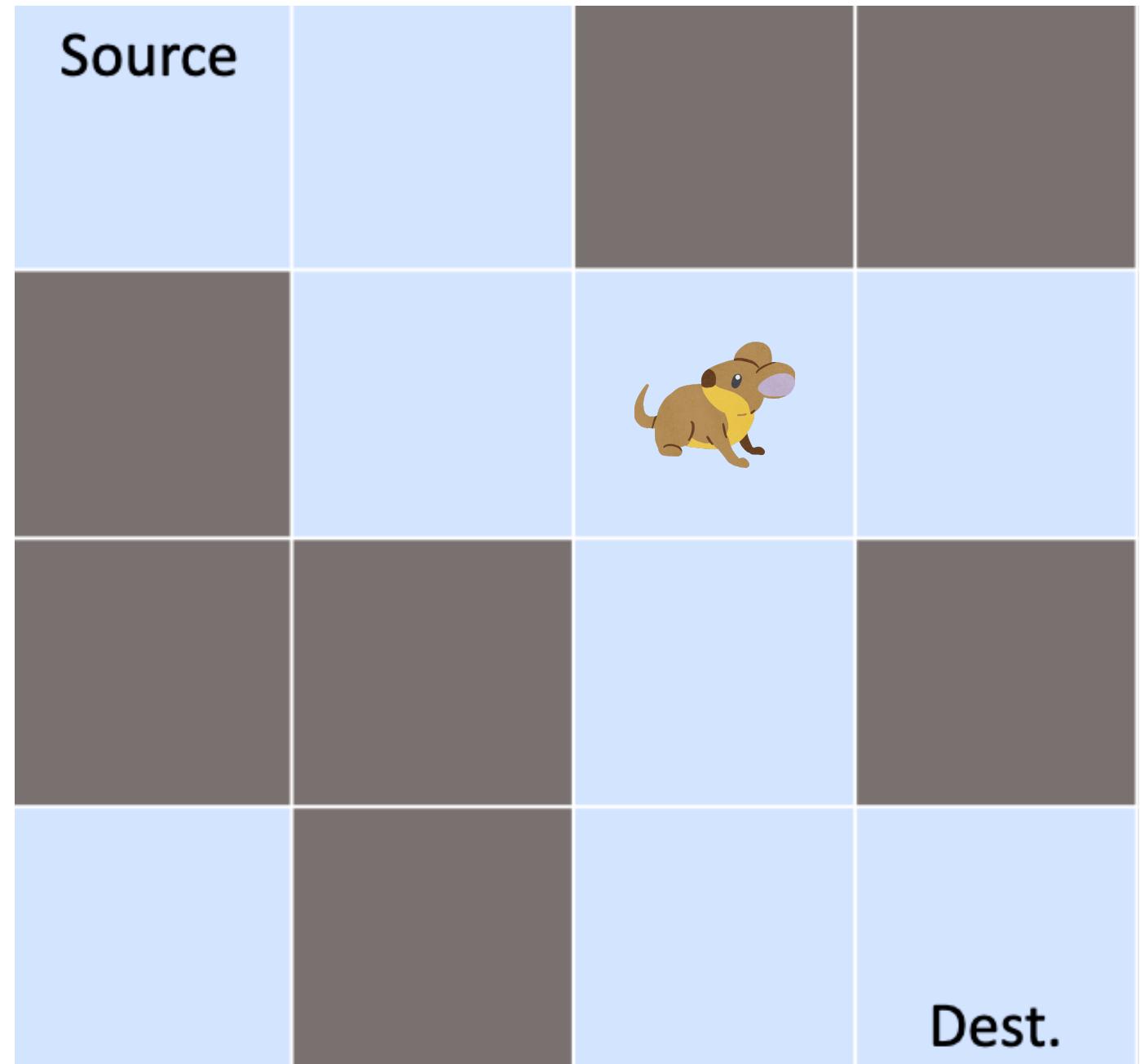
Rat in a maze



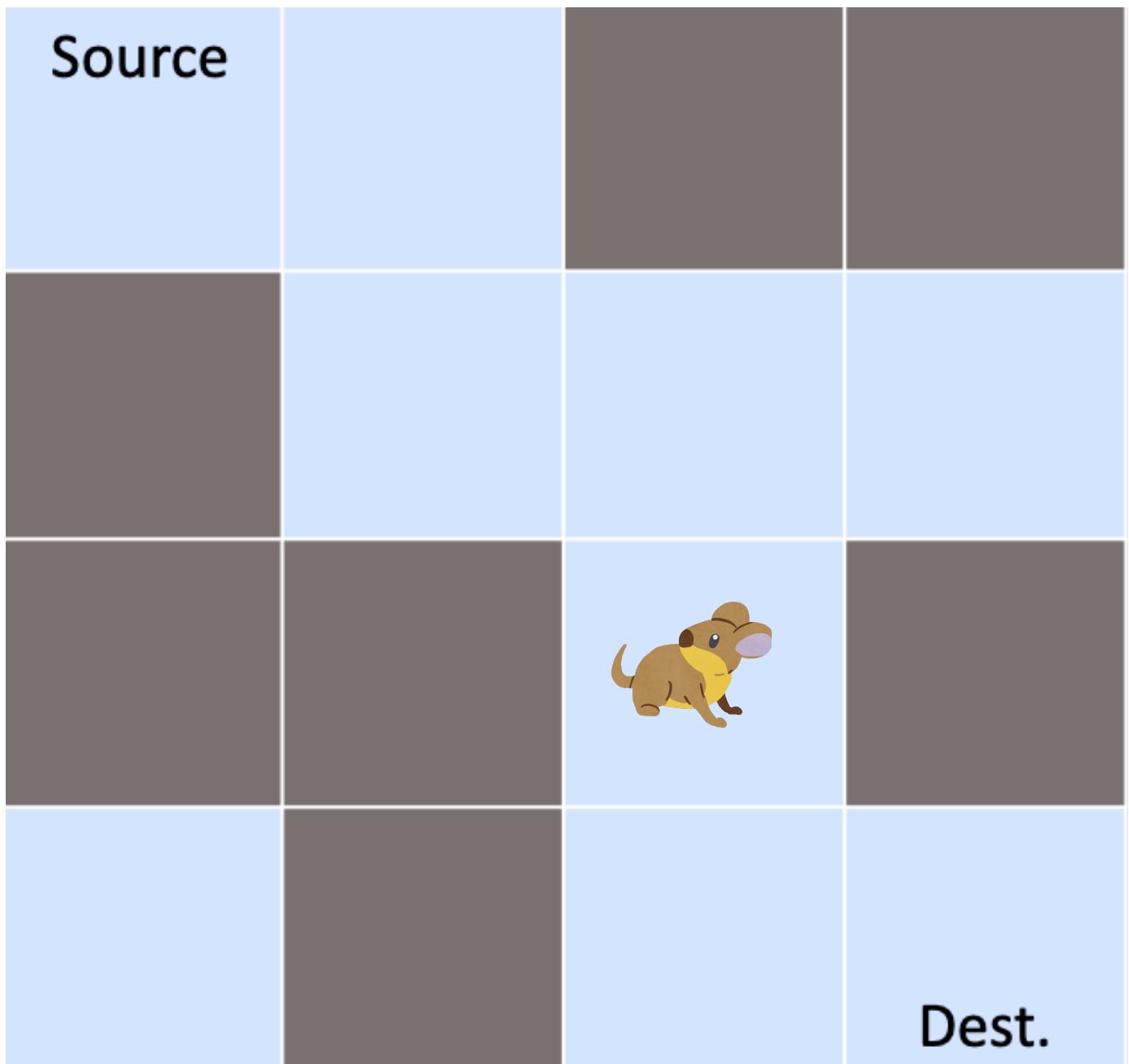
Rat in a maze



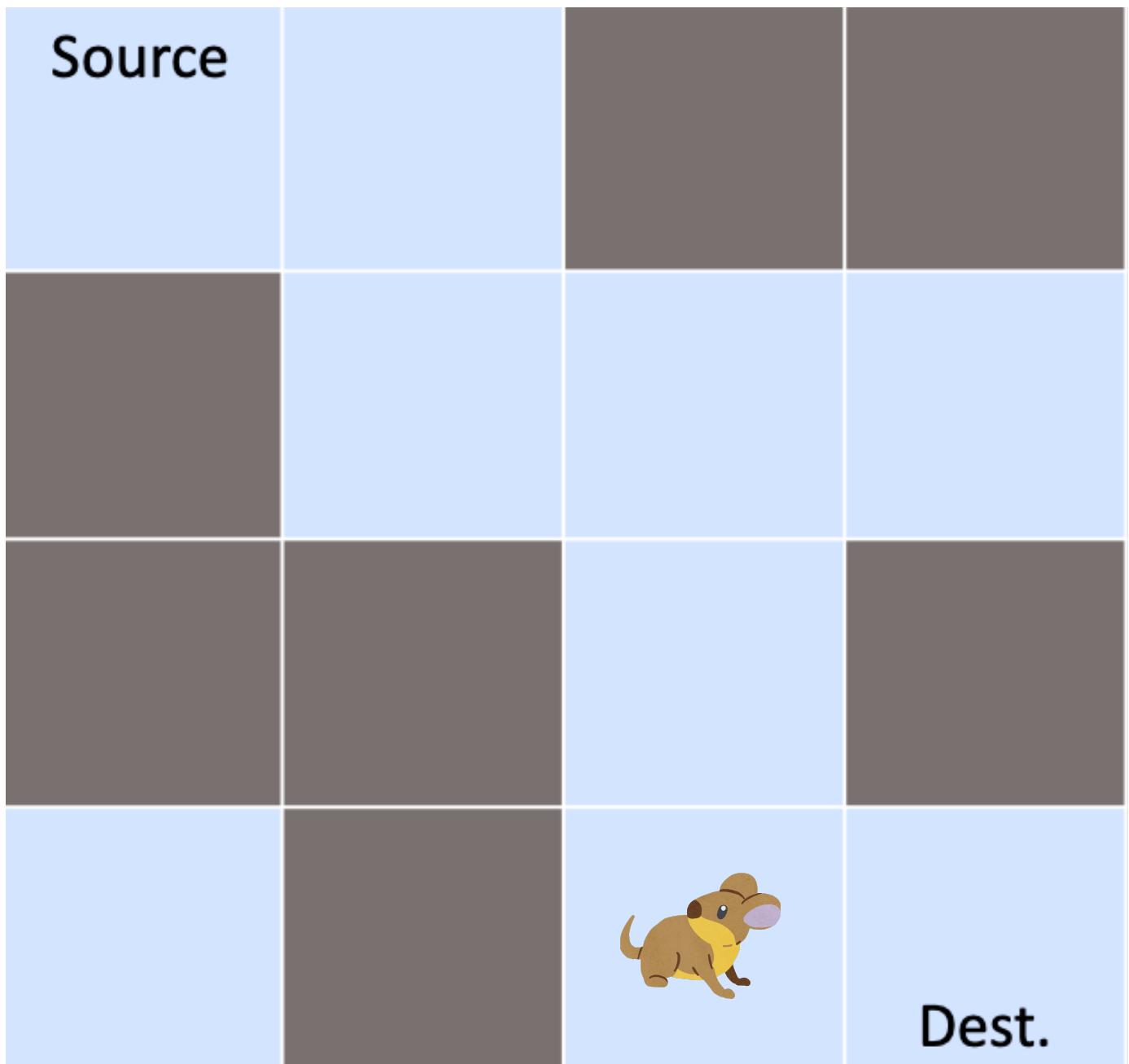
Rat in a maze



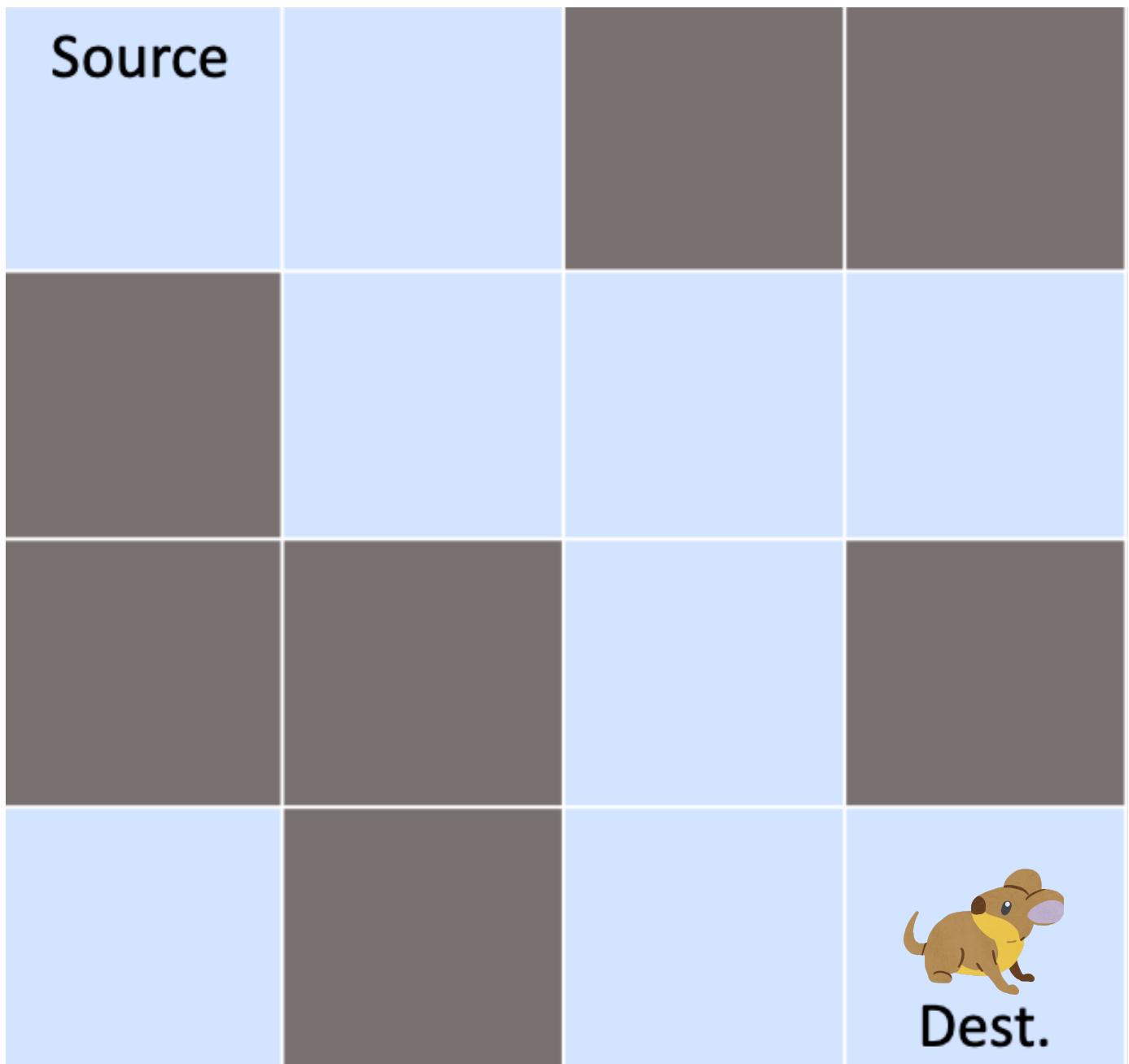
Rat in a maze



Rat in a maze



Rat in a maze



Sudoku



Constraints

- Each row must have unique numbers from 1-9
- Each column has unique numbers from 1-9
- Each sub-grid has unique numbers from 1-9

3		6	5		8	4		
5	2							
	8	7					3	1
		3		1			8	
9			8	6	3			5
	5			9		6		
1	3				2	5		
					7	4		
	5	2		6	3			



Sudoku

3	1	6	5		8	4		
5	2							
	8	7				3	1	
		3	1			8		
9		8	6	3			5	
	5		9		6			
1	3				2	5		
						7	4	
	5	2		6	3			

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	2	8	4		
5	2							
	8	7				3	1	
		3		1		8		
9			8	6	3			5
	5			9		6		
1	3				2	5		
						7	4	
		5	2		6	3		

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	2	8	4	9	7
5	2	4	1	3	7	8	6	
	8	7					3	1
		3		1			8	
9			8	6	3			5
	5			9		6		
1	3					2	5	
						7	4	
	5	2		6	3			

No solution?

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	2	8	4	9	7
5	2	4	1	3	7	8		
	8	7					3	1
		3		1			8	
9			8	6	3			5
	5			9		6		
1	3					2	5	
							7	4
		5	2		6	3		

No solution?

Backtrack

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	2	8	4	9	7
5	2	4	1	3	7			
	8	7				3	1	
		3		1		8		
9			8	6	3			5
	5			9		6		
1	3				2	5		
						7	4	
		5	2		6	3		

No solution?

Backtrack

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	2	8	4	9	7
5	2	4	1	3	9			
	8	7				3	1	
		3		1		8		
9			8	6	3			5
	5			9		6		
1	3				2	5		
						7	4	
		5	2		6	3		

No solution?

Backtrack

Retry

The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Sudoku

3	1	6	5	7	8	4	9	2
5	2	9	1	3	4	7	6	8
4	8	7	6	2	9	5	3	1
2	6	3	4	1	5	9	8	7
9	7	4	8	6	3	1	2	5
8	5	1	7	9	2	6	4	3
1	3	8	9	4	7	2	5	6
6	9	2	3	5	1	8	7	4
7	4	5	2	8	6	3	1	9

No solution?

Backtrack

Retry

Found Solution!

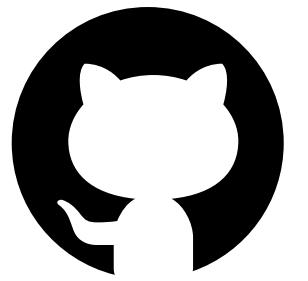
The backtracking algorithm here will try to place each number in each row and column following the constraints until the puzzle is solved.



Code



Frameworks:
Tkinter



Link to the project files

Modules:
Time
Math
Random
Datetime
PIL
Numpy
Copy
Ctypes



Project Snapshots



Solving Constraint Satisfaction Problems using RAT

Varun C
Vishal M Godi
Sarthak S Kumar
Sem 1 CSE, PESU ECC

Welcome

Select one to continue



Solve Sudoku



Rat in a Maze

Exit

main.py





Solve Sudoku

Sudoku $\frac{1}{2}$

Welcome

Your Name: PESU

Difficulty: Intermediate

Submit

9	7	5	1	6	3	8	2	4
6	2	1	4	8	9	3	5	7
4	3	8	5	2	7	9	1	6
5	4	2	7	9	1	6	3	8
1	8	7	6	3	4	5	9	2
3	9	6	2	5	8	4	7	1
2	1	9	3	4	6	7	8	5
8	5	4	9	7	2	1	6	3
7	6	3	8	1	5	2	4	9

✗ You gave up so quick!

Better luck next time

With Experience comes Expertise!

Next

	5		6	3	8	2		
	2	1		9	3	5		
	8		2			6		
5	4			1	6	8		
1		6		4	5	9	2	
3	9			8	4	7	1	
	9		6		8			
5	4	9	7		1	3		
7	6		1	5	2	4	9	

No relief mate!

Check it again!

Chances Remaining: 9

Check Solution

Solve Sudoku

	5		6	3	8	2		
	2	1		9	3	5		
	8		2			6		
5	4			1	6	8		
1		6		4	5	9	2	
3	9			8	4	7	1	
	9		6		8			
5	4	9	7		1	3		
7	6		1	5	2	4	9	

Hello PESU
Your sudoku is here. Go ahead, Solve it

Use to enter numbers
and to move the selector

Check Solution

Solve Sudoku

Solving Sudoku

Wanna solve another maze?

Yeah Nah

Click!

Better luck next time

With Experience comes Expertise!

Next

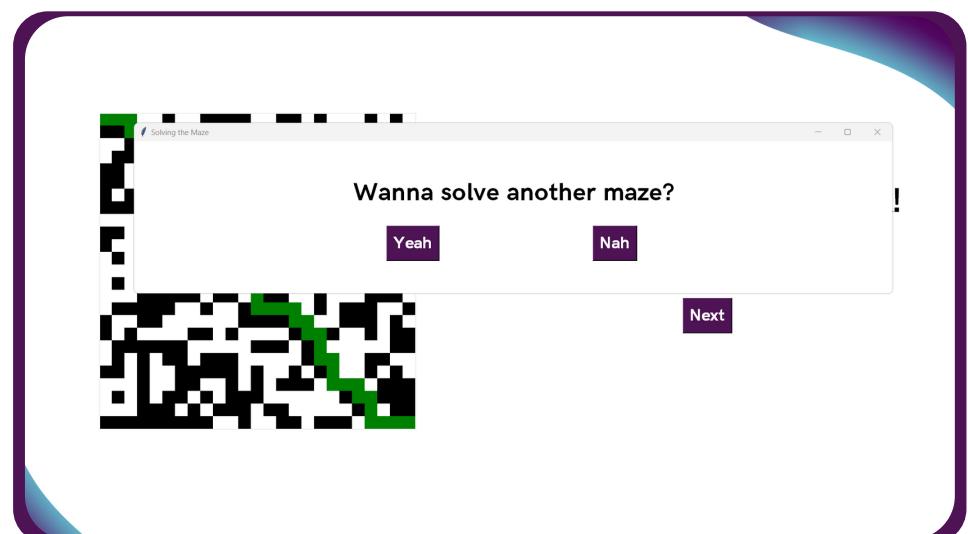
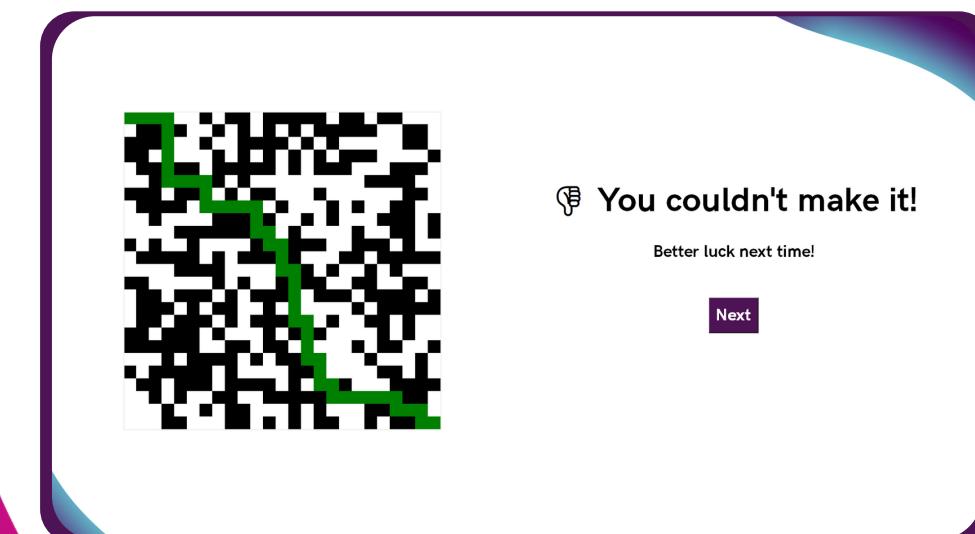
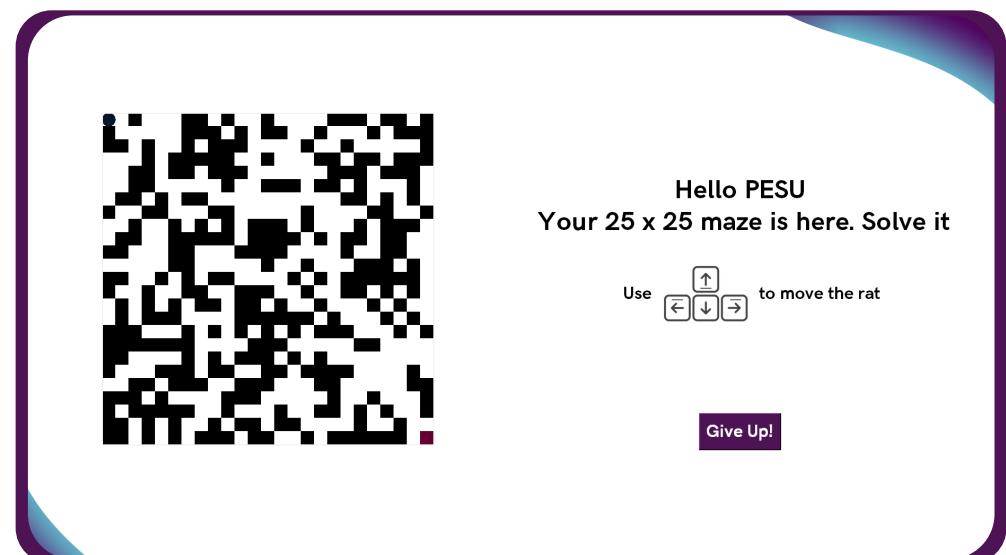
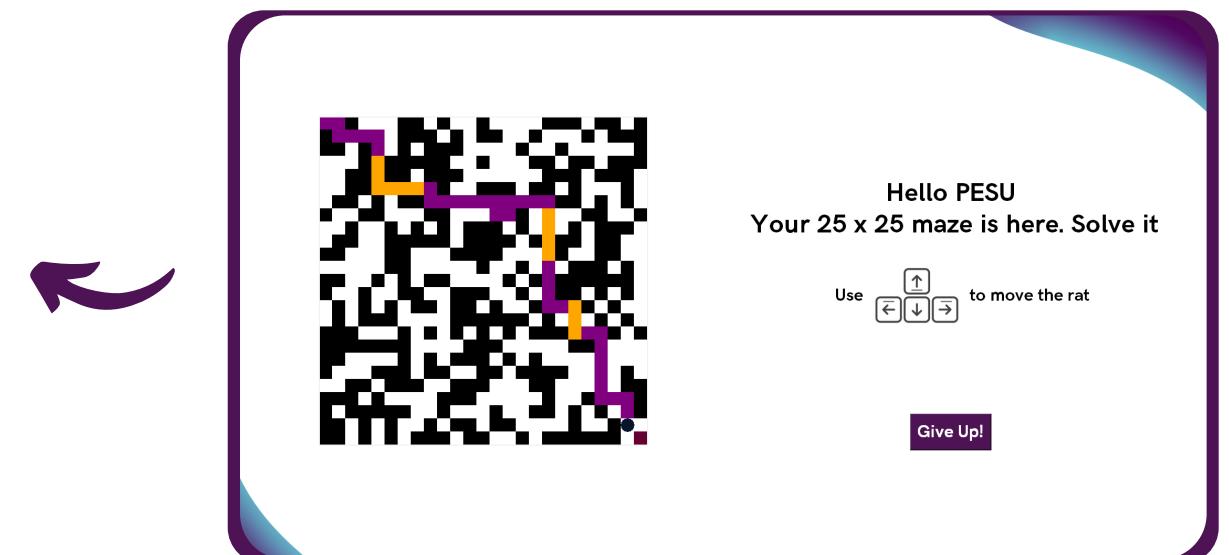
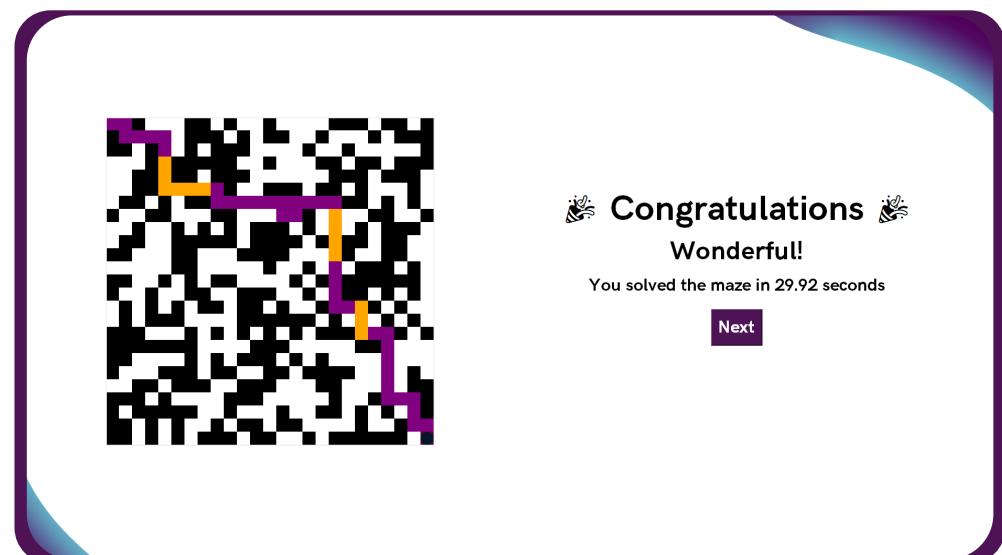
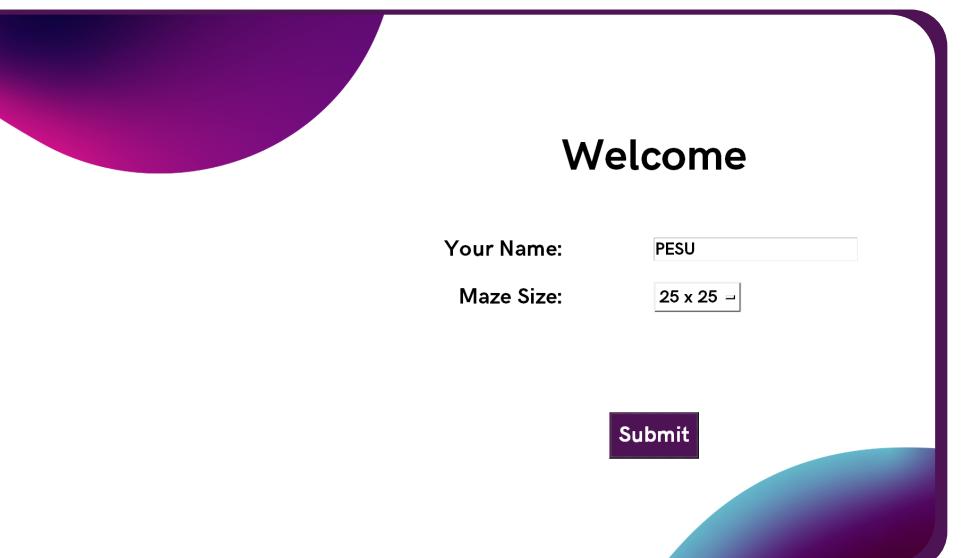
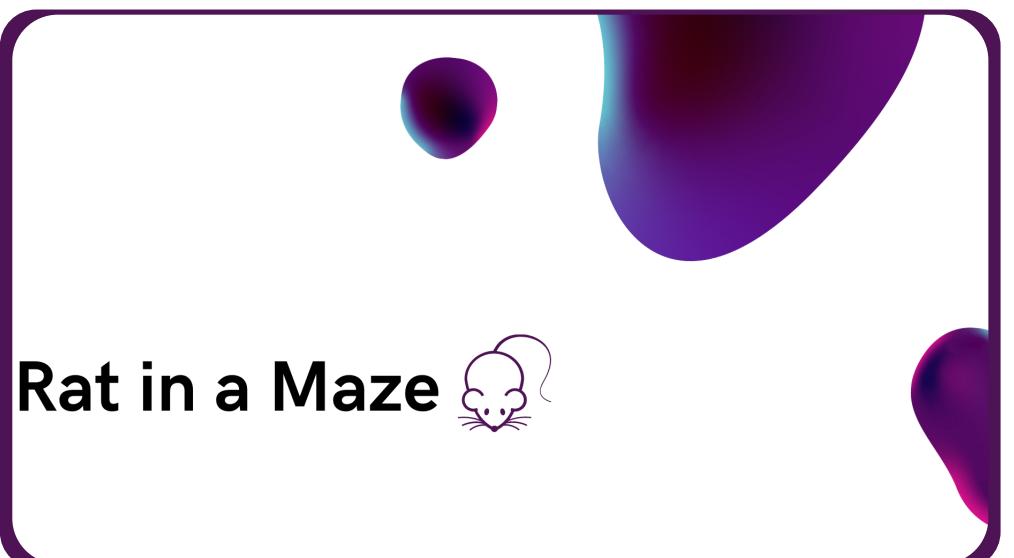
5	4	2	7	9	1	6	3	8
1	8	7	6	3	4	5	9	2
3	9	6	2	5	8	4	7	1
2	1	9	3	4	6	7	8	5
8	5	4	9	7	2	1	6	3
7	6	3	8	1	5	2	4	9

solve_sudoku_UI.py





Rat in a Maze



Rat_in_a_maze_UI.py



Thank you!

Solving constraint satisfaction problems using RAT

