**CS657 Assignment 1 Report**

Dong, Jiang (819364904)

**INTRODUCTION:**

This project includes 5 parts: Position, Maze, Orientation, Visualization and Robot. Robot is the main part where defines the rules to find the destination. Visualization is used to show the result of the whole process, in which the red color indicates the obstacle; the yellow color indicates the path the robot already detected; the blue color indicates the path the robot found to go to the destination.

**RULES:**

1) **Rule1:**
   If destination is in the front path (the robot can see destination in the front), then go straightforward until arrived the destination.
2) **Rule2:**
   If destination is in the right front path (the robot can see destination in right front), then turn 45 degrees to the right and go straightforward until arrived the destination.
3) **Rule3:**
   If destination is in the left front path (the robot can see destination in the left front), then turn 45 degree to the left and go straightforward until arrived the destination.
4) **Rule4:**
   If the robot can ‘t see destination and the front path is not blocked, then go straightforward.
5) **Rule5:**
   If the robot can ‘t see destination and the front path is blocked, if left front path is not blocked then move one step to left front.
6) **Rule6:**
   If the robot can ‘t see destination and the front path, left front path are all blocked then move one step to right front.
7) **Rule7:**
   If the robot can’t see destination and all the 3 paths are blocked, then turn 45 degree to the left and detect the path again.

**PROGRAM FEATURES:**

1) **DetectPath():**
   Every time when the robot arrives at a position, he will detect the path in 3 directions: front, left front and right front. Then the robot will update which cells are available.

2) **frontPath():**
   Sub-function of DetectPath(), used to detect the front path of the maze.

3) **rightFrontPath()**
   Sub-function of DetectPath(), used to detect the right front path of the maze.

4) **leftFrontPath():**
   Sub-function of DetectPath(), used to detect the left front path of the maze.

5) **MoveForward():**
   This action is based on the current orientation of the robot. There are totally 8 orientations, the robot will update the position based on the current orientation. For example, if the current orientation is north, after executing MoveForward(), the coordinate of y will plus one. After updated the current position, it will also be recorded in pathRecorder().

6) **MoveRight():**
   Similar to MoveForward(), the only difference is the robot will turn 45 degree to the right first then update he current position based on the current orientation. After that, pathRecorder() will record this move.

7) **MoveLeft():**
   The only difference to MoveRight() is turn 45 degree to the left.

8) **turnLeft()**
   The robot will turn 45 degrees to the left. The current orientation will be updated (minus one)

9) **turnRight()**
   The robot will turn 45 degrees to the right. The current orientation will be updated (plus one)

10) **pathRecorder**
    pathRecorder is a stack used to record the path of the robot.

11) **goBack()**
    The robot will go back to the previous position if needed. The top position in the stack will be popped and the current position of the robot will be set to the current top position of the stack.

**EXPERIMENTS RUN:**

**Experiment1:**

Settings: (Modify the parameter in constructor of the robot to change settings.)
Obstacle percentage: 10%
Initial orientation: North
Start position: (5,5)
Destination: (29,29)

**Experiment2:**

Settings: (Modify the parameter in constructor of the robot to change settings.)
Obstacle percentage: 25%
Initial orientation: North
Start position: (5,5)
Destination: (29,29)

**Experiment3:**

**EXPERIMENTAL RESULT:**

**Experiment1:**

As shown in Fig 1, red indicates obstacles, white indicates the undetected area of the maze, yellow indicates the detected area by the robot, blue indicates the path from starting position to the destination. There are **118** steps to go to the destination. Here below are all the positions in the path:

**Robot path(118 steps):**

Path is 5 , 5
Path is 5 , 6
Path is 5 , 7
Path is 5 , 8
Path is 5 , 9
Path is 5 , 10
Path is 5 , 11
Path is 5 , 12
Path is 6 , 13
Path is 7 , 14
Path is 8 , 15
Path is 9 , 16
Path is 10 , 17
Path is 11 , 18
Path is 12 , 19
Path is 13 , 20
Path is 14 , 21

Path is 15 , 21
Path is 16 , 20
Path is 17 , 19
Path is 18 , 18
Path is 19 , 17
Path is 20 , 16
Path is 21 , 15
Path is 22 , 14
Path is 23 , 13
Path is 24 , 12
Path is 24 , 11
Path is 24 , 10
Path is 24 , 9
Path is 24 , 8
Path is 24 , 7
Path is 24 , 6
Path is 24 , 5
Path is 24 , 4
Path is 24 , 3
Path is 24 , 2
Path is 24 , 1
Path is 24 , 0
Path is 23 , 0
Path is 22 , 0
Path is 21 , 0
Path is 20 , 0
Path is 19 , 0
Path is 18 , 0
Path is 17 , 0
Path is 16 , 0
Path is 15 , 0
Path is 14 , 1
Path is 14 , 2
Path is 14 , 3
Path is 14 , 4
Path is 15 , 5
Path is 16 , 6
Path is 17 , 7
Path is 18 , 8
Path is 19 , 9
Path is 20 , 10

Path is 21 , 10
Path is 22 , 10
Path is 23 , 10
Path is 24 , 10
Path is 25 , 10
Path is 26 , 10
Path is 27 , 10
Path is 28 , 10
Path is 29 , 10
Path is 29 , 9
Path is 29 , 8
Path is 29 , 7
Path is 29 , 6
Path is 28 , 7
Path is 27 , 8
Path is 26 , 9
Path is 25 , 10
Path is 24 , 11
Path is 23 , 12
Path is 22 , 13
Path is 21 , 14
Path is 20 , 15
Path is 19 , 16
Path is 18 , 17
Path is 17 , 18
Path is 17 , 19
Path is 17 , 20
Path is 17 , 21
Path is 17 , 22
Path is 17 , 23
Path is 18 , 24
Path is 19 , 25
Path is 20 , 26
Path is 21 , 27
Path is 22 , 27
Path is 23 , 27
Path is 24 , 27
Path is 25 , 27
Path is 26 , 27
Path is 27 , 27
Path is 28 , 28

Path is 29 , 29

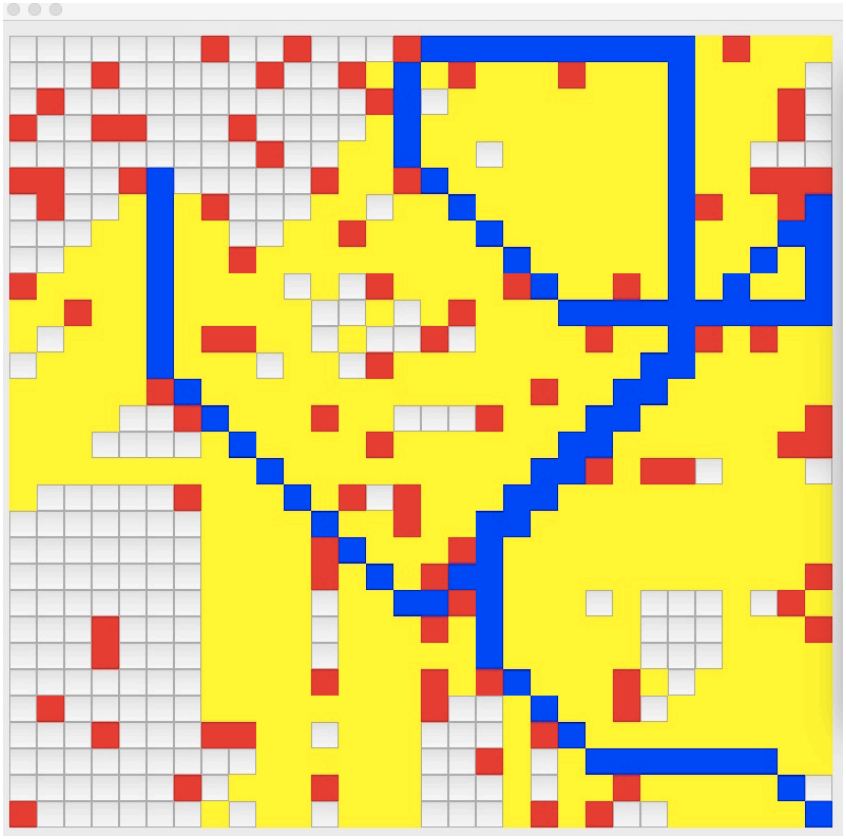

Fig 1. Experimental run result1 10% obstacle
(Red: obstacles, White: undetected area, Yellow: detected area, Blue: robot path)

**Experiment2:**
As shown in Fig 2, the blue one is the path of the robot, this time there are **222** steps to the destination.

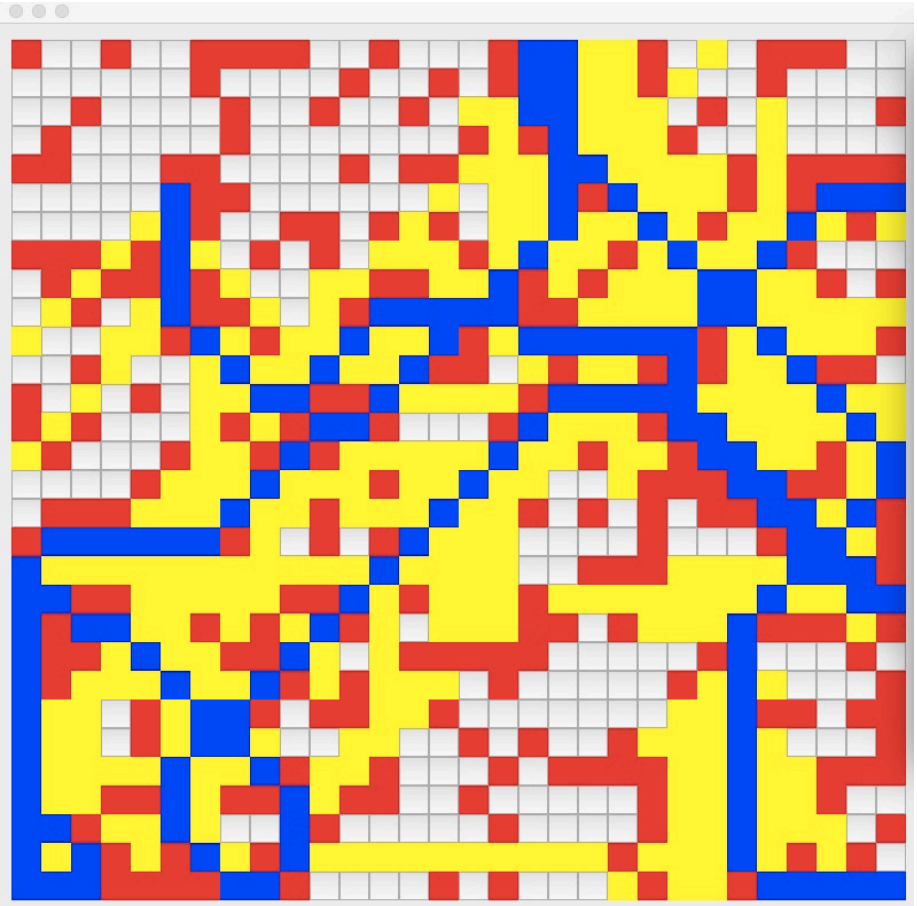

Fig 2. Experimental run result2 25% obstacle

(Red: obstacles, White: undetected area, Yellow: detected area, Blue: robot path)

**Robot path(222steps):**

Path is 5 , 5
Path is 5 , 6
Path is 5 , 7
Path is 5 , 8
Path is 5 , 9
Path is 6 , 10
Path is 7 , 11
Path is 8 , 12
Path is 9 , 12
Path is 10 , 11
Path is 11 , 10
Path is 12 , 9
Path is 13 , 9
Path is 14 , 9
Path is 15 , 9

Path is 16 , 9
Path is 17 , 10
Path is 18 , 10
Path is 19 , 10
Path is 20 , 10
Path is 21 , 10
Path is 22 , 10
Path is 23 , 9
Path is 24 , 8
Path is 25 , 7
Path is 26 , 6
Path is 27 , 5
Path is 28 , 5
Path is 29 , 5
Path is 28 , 5
Path is 27 , 5
Path is 26 , 6
Path is 25 , 7
Path is 24 , 8
Path is 23 , 9
Path is 22 , 10
Path is 22 , 11
Path is 22 , 12
Path is 22 , 13
Path is 23 , 14
Path is 24 , 15
Path is 25 , 16
Path is 26 , 17
Path is 27 , 18
Path is 28 , 19
Path is 29 , 19
Path is 28 , 18
Path is 27 , 17
Path is 26 , 16
Path is 25 , 15
Path is 24 , 14
Path is 23 , 13
Path is 22 , 12
Path is 21 , 12
Path is 20 , 12
Path is 19 , 12

Path is 18 , 12
Path is 17 , 13
Path is 16 , 14
Path is 15 , 15
Path is 14 , 16
Path is 13 , 17
Path is 12 , 18
Path is 11 , 19
Path is 10 , 20
Path is 9 , 21
Path is 8 , 22
Path is 7 , 23
Path is 6 , 24
Path is 5 , 25
Path is 5 , 26
Path is 5 , 27
Path is 6 , 28
Path is 7 , 29
Path is 8 , 29
Path is 9 , 28
Path is 9 , 27
Path is 9 , 26
Path is 8 , 25
Path is 7 , 24
Path is 6 , 23
Path is 5 , 22
Path is 4 , 21
Path is 3 , 20
Path is 2 , 20
Path is 1 , 19
Path is 0 , 18
Path is 0 , 19
Path is 0 , 20
Path is 0 , 21
Path is 0 , 22
Path is 0 , 23
Path is 0 , 24
Path is 0 , 25
Path is 0 , 26
Path is 0 , 27
Path is 0 , 28

Path is 0 , 29
Path is 1 , 29
Path is 2 , 29
Path is 2 , 28
Path is 1 , 27
Path is 0 , 26
Path is 0 , 25
Path is 0 , 24
Path is 0 , 23
Path is 0 , 22
Path is 0 , 21
Path is 0 , 20
Path is 0 , 19
Path is 0 , 18
Path is 1 , 17
Path is 2 , 17
Path is 3 , 17
Path is 4 , 17
Path is 5 , 17
Path is 6 , 17
Path is 7 , 16
Path is 8 , 15
Path is 9 , 14
Path is 10 , 13
Path is 11 , 13
Path is 12 , 12
Path is 13 , 11
Path is 14 , 10
Path is 15 , 9
Path is 16 , 8
Path is 17 , 7
Path is 18 , 6
Path is 18 , 5
Path is 18 , 4
Path is 18 , 3
Path is 18 , 2
Path is 18 , 1
Path is 18 , 0
Path is 17 , 0
Path is 17 , 1
Path is 17 , 2

Path is 18 , 3
Path is 19 , 4
Path is 20 , 5
Path is 21 , 6
Path is 22 , 7
Path is 23 , 8
Path is 24 , 9
Path is 25 , 10
Path is 26 , 11
Path is 27 , 12
Path is 28 , 13
Path is 29 , 14
Path is 29 , 15
Path is 28 , 16
Path is 27 , 17
Path is 26 , 18
Path is 25 , 19
Path is 24 , 20
Path is 24 , 21
Path is 24 , 22
Path is 24 , 23
Path is 24 , 24
Path is 24 , 25
Path is 24 , 26
Path is 24 , 27
Path is 24 , 28
Path is 25 , 29
Path is 26 , 29
Path is 27 , 29
Path is 28 , 29
Path is 29 , 29

**DISCUSSION OF RESULT:**

Based on the rules I designed, the robot can find a solution to the destination, but obviously this is not the shortest path. In both cases, the robot should go all the way down to the right lower corner, no need to go to the right upper corner and then go back, so the rules still need to be optimized.

**CONCLUSIONS:**

Since the robot have no idea about the whole picture of the maze in advance, it is impossible to get the optimized solution for minimal steps. However, the rules I

designed still need to be optimized, for example, if the robot has multiple paths available, it should select the path that has the nearest distance to the destination, etc.
All in all, this solution can be used to find the destination but still need to be optimized for fewer steps to the destination.