

## Square Room

#Approach: The first step was to send Picobot to the top right corner. Once the Picobot was in the right position, send it in a loop that would guide it South as far as possible -> East one unit -> North as far as possible -> East one unit -> repeat until it completes the coverage of the room.

# state 0 goes to the top of the room and directs the bot to go west on state 1

0 x\*\*\* -> N 0 # if there is nothing to North, go North

0 N\*x\* -> W 1 # if North is blocked and West is free, go West and switch to state 1

0 N\*W\* -> X 1 # if North and West are blocked, switch to state 1

# state 1 goes West as far as possible. Since the bot was on the top of the square, this state goes to the top left corner and directs the bot to go south on state 2

1 \*\*x\* -> W 1 # if there is nothing to West, go West

1 \*\*Wx -> S 2 # if West is blocked and South is free, go south and switch to state 2

# state 2 goes South as far as possible then one unit East. It switches to state 3

2 \*\*\*x -> S 2 # if there is nothing South, go South

2 \*x\*S -> E 3 # if East is free and South is blocked, go East one unit and switch to state 3

# state 3 goes North as far as possible then one unit East. It switches to state 2 so it creates a loop that will keep repeating itself until the room is filled and the picobot is at the bottom right corner.

3 x\*\*\* -> N 3 # if there is nothing North, go North

3 N\*\*\* -> E 2 # if North is blocked, go East and switch to state 2

## Diamond Room

#Approach: The first step was to send the Picobot to the extreme left side from which I would have to move only right to complete half of the diamond. Once Picobot was at the desired position, the idea was to send it into a loop, (moving North all the way -> South all the way -> one unit East -> repeat) that would fill the left half of the diamond. Next I followed the same strategy for the right side. Send Picobot to the extreme right of the map, send it into a loop, (moving North all the way -> South all the way -> one unit West -> repeat) until it completed the whole map.

# state 0 has the objective of getting Picobot to the left extreme of the diamond and preparing it for state 1. Wherever Picobot is, state 0 will take it to the left wall and proceed down or up the “stairs” until it reaches the point where only East is free.

0 \*\*x\* -> W 0 # if West is free, go West

0 x\*WS -> N 0 # if North is free and West and South are blocked, move North

0 N\*Wx -> S 0 # if North and West are blocked and South is free, go south

0 NxWS -> E 1 # if North, West and South are blocked and only East is free, move East (preparing for the next state) and switch to state 1

# state 1 goes North as far as possible and then one unit South, preparing it for state 2

1 xxxx -> N 1 # if all the sides are free, move North as far as possible

1 N\*\*x -> S 2 # if North is blocked and South is free, move one unit South and switch to state 2

# state 2 goes as South as possible until it hits a wall on the South side, then it moves one unit East and switches back to state 1. By doing so, it creates a loop that will run until the left side of the diamond is completely filled. Once the left side is filled, the Picobot will hit the extreme South in which the last rule will be applied, switching to state 3, that will work with the right side of the diamond.

2 xxxx -> S 2 # if all the sides are free, move South as far as possible

2 \*x\*S -> E 1 # if there is nothing to East and South is blocked, move East one unit. Switch to state 1. By switching to state 1, picobot will enter a loop.

2 xEWS -> N 3 # if North is free and East, West and South are blocked, move one unit North. Switch to state 3. When it arrives at this point, Picobot will exit the loop and move to state 3.

# state 3 has the objective of getting Picobot to the right extreme of the diamond and preparing it for state 4. Wherever Picobot is, state 3 will take it to the right wall and proceed down or up the “stairs” until it reaches the point where only West is free.

3 xxxx -> E 3 # if all the sides are free, move East

3 xExS -> N 3 # if North and West are free, and East and South are blocked, move North

3 NExS -> W 4 # if North, East and South are blocked, and West is free, move West. Switch to state 4.

# state 4 goes North as far as possible and then one unit South, preparing it for state 5 and a loop

4 xxxx -> N 4 # if all the sides are free move North as far as possible

4 N\*\*x -> S 5 # if North is blocked and South is free, move one unit South and switch to state 5

# state 5 goes as South as possible until it hits a wall on the South side, then it moves one unit West and switches back to state 4. By doing so, it creates a loop that will run until the diamond is completely filled.

5 xxxx -> S 5 # if all the sides are free, move South as far as possible

5 \*\*xS -> W 4 # if there is nothing to West and South is blocked, move West one unit. Switch to state 4. By switching to state 4, picobot will enter a loop.

## Columns Room

#Approach: First step was to send Picobot to the North-western corner of the room, so from there, using a pattern that would cause a loop, make Picobot cover the most part of the room (south of every obstacle) by moving North all the way -> South all the way -> East one unit. Once Picobot had covered the south of every obstacle, the idea was to cover the North blank spaces by positioning the Picobot at the top right corner and move it West all the way -> South one unit -> East all the way -> repeat until it hit the 2x2 square. Once that was done, covering the blank space under the picobot was the easiest move by using the North, South, East pattern. The next step was to fill in the white line above the 1x1 square. Positioning the Picobot over the square and moving North solved it. Following that, the objective was to cover the white space above the 3x3 square, what could be done by using a West, South, East, South pattern. Once it hit the "I" shaped obstacle, adjust the coding so it would fill the right side of it. At last, fill in the left side of the "I" and the remaining white spaces above the 3x3 square, always on the East, west motion.

# state 0 has the objective of getting Picobot to the top left corner

0 \*\*X\* -> W 0

0 x\*W\* -> N 0

0 NxWx -> E 1

# state 1 has the same objective of state 0 (get picobot to the top left corner), but it considers possible obstacles on the way. If the path is clear, state 1 will make little to no difference, making sure state 2 still works

1 xxxx -> N 1

1 Nxxx -> W 1

1 xxWx -> N 1

1 NxWx -> S 2

# state 2 is responsible for the first part of the loop, making Picobot go all the way South and one unit East

2 \*\*\*X -> S 2

2 \*X\*S -> E 3

# state 3 is responsible for the second part of the loop, making sure picobot stays in the loop for the necessary time and assuring it will move North all the way and one unit South. When Picobot gets to the top left corner, it exits the loop

3 X\*\*\* -> N 3

3 Nx\*\* -> S 2  
3 NExx -> W 4

# state 4 covers some of the white parts on the north side of the map, simulating a mini loop.

4 N\*x\* -> W 4  
4 NxWx -> S 4  
4 xxWx -> E 4  
4 xxx\* -> E 4  
4 xExx -> W 5

# state 5 is a continuation of step 4 with the East - West motion

5 xxx\* -> W 5  
5 \*\*Wx -> S 6

# state 6 is the last step completing the objective of the East - West motion to fill up the North side of the map

6 xx\*\* -> E 6  
6 xExx -> S 7

# state 7,8,9 work on a loop with the North-South motion to fill the blank space under the 2x2 square. They are also responsible for taking Picobot to the bottom right corner of the map.

7 xExx -> W 7  
7 xxxx -> S 7  
7 xxxS -> E 8

8 xxxS -> N 8  
8 x\*\*x -> N 8  
8 N\*\*\* -> S 9  
8 xExS -> W 10

9 x\*\*x -> S 9  
9 xxxS -> E 8

# states 10, 12, 13, 14, 15 are responsible for a loop that will take the picobot in a East-West motion to the top of the 1x1 square.

10 xxxS -> W 10  
10 xxWS -> N 10  
10 \*xWx -> E 10  
10 \*xxx -> E 10

10 xExS -> W 11  
10 \*E\*x -> W 12

# state 11 works with the idea that if there is an obstacle it will get the picobot around it.

11 \*xxx -> W 11  
11 \*xWx -> N 10

12 xxxx -> W 12  
12 xxWx -> N 13

13 xxWx -> E 13  
13 \*xxx -> E 13  
13 xExx -> N 14

14 xExx -> N 15

15 xExx -> W 15  
15 \*xxx -> W 15  
15 xxxS -> N 16

# state 16 is responsible for covering the white space above the 1x1 square

16 xxxx -> N 16  
16 Nxxx -> W 17

# state 17,18,19,20 are responsible for covering the part of the white spot on top of the 3x3 square and the right side of the "I" shaped obstacle using a East-West loop.

17 Nxxx -> W 17  
17 xxxx -> S 18

18 xxx\* -> E 18  
18 xExx -> S 19  
18 Nxxx -> W 21

19 xExx -> W 19  
19 xxxx -> W 19  
19 xxWx -> S 20

20 xxWx -> E 18  
20 \*xx\* -> W 20

20 NxWx -> S 20

20 xxWS -> E 18

# states 21, 22 are responsible for the left side of the “I” shaped obstacle

21 xxxx -> N 21

21 xExx -> N 22

22 xExx -> N 22

22 xxxx -> E 22

22 xExS -> N 22

22 NExx -> W 23

# state 23 takes picobot to the 3x3 square, which is the closest obstacle to the white zone.

23 xxxx -> W 23

23 xxWx -> N 23

23 xxxS -> W 24

# states 24, 25, 26 take care of the remaining area above the 3x3 square with a East-West motion.

24 xxxS -> W 24

24 xxxx -> N 25

25 xxxx -> E 25

25 xExx -> N 26

26 xExx -> W 26

26 xxxx -> W 26