**Picobot, Room 1**

#This state 0 will lead PicoBot to the bottom left corner of the room; it will not be utilized again.

0 \*\*\*x -> S 0 # Keep moving south

0 \*\*xS -> W 0 # Head toward the southwest corner

0 \*\*WS -> x 1 # Reached the corner! Now to move up and down.

1 x\*\*\* -> N 1 #PicoBot will follow a N-S pattern.

1 Nx\*\* -> E 2 #Move one step East in preparation to go South

1 NE\*\* -> x 3 # If N and E are blocked, it's an opposite corner. Done!

2 \*\*\*x -> S 2 # Start to move south!

2 \*x\*S -> E 1 # Prepare for the next journey up north.

2 \*E\*S -> x 3 # If S and E are blocked, it's an opposite corner. Done!

3 \*\*\*\* -> x 3 #If it's done, it sits.

**Picobot, Maze**

#Basically, what we are trying to do is use the right-wall approach to figure out how to navigate the maze. The robot follows its right wall; if it loses sight of the right wall at any point, it will turn right (like at an intersection) to find the right wall again. We start by testing East - it moves right, following its right wall.

0 \*x\*\* -> E 3 #Move E if East is not blocked.

0 xE\*\* -> N 0 #Move N if East is blocked.

0 NEx\* -> W 1 #Move W if N & E are both blocked.

0 NEWx -> S 2 #If all else fails (i.e. it's in a dead end), move South.

#Now we start by testing North; if East is blocked, then PicoBot turns left to go to North (the next right wall to follow).

1 x\*\*\* -> N 0 #testing all the possible directions

1 N\*x\* -> W 1

1 N\*Wx -> S 2

1 NxWS -> E 3

#Now we test West; if North/East is blocked, then the wall will lead the robot west.

2 \*\*x\* -> W 1 #testing all possible directions

2 \*\*Wx -> S 2

2 \*xWS -> E 3

2 xEWS -> N 0

#Finally, we test South; if N, E, and W are blocked, then PicoBot can only move south, along the right wall.

3 \*\*\*x -> S 2 #testing all possible directions

3 \*x\*S -> E 3

3 xE\*S -> N 0

3 NExS -> W 1