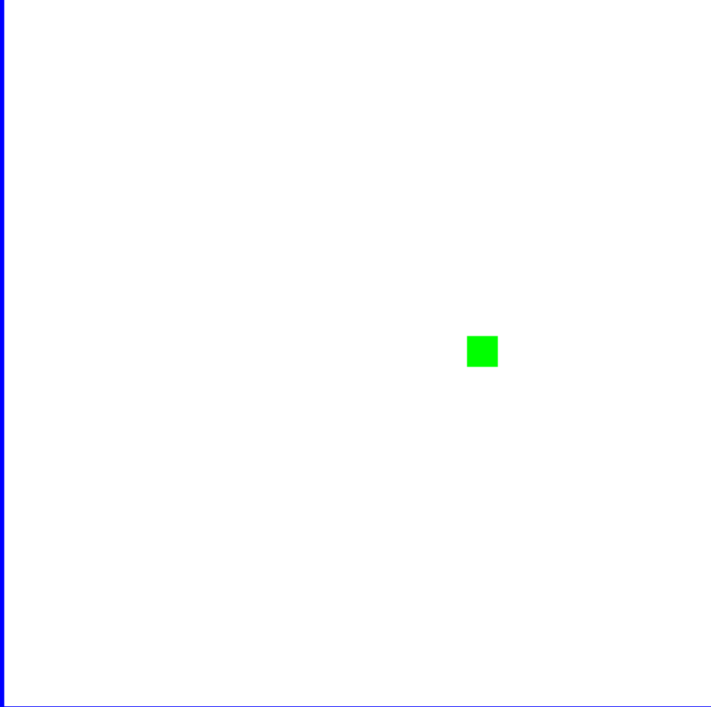


# Picobot

Based on lecture slides from Harvey Mudd College



## Picobot

### Rules

```

# These lines are comments.

# Remember that rules are formatted as
# State Surroundings -> Move NewState

# Picobot starts in state 0.
# Here, state 0 goes N as far as possible

0 x** -> N 0 # if there's nothing to the N, go N
0 N** -> X 1 # if N is blocked, switch to state 1

# and state 1 goes S as far as possible

1 *** -> S 1 # if there's nothing to the S, go S
1 ***S -> X 0 # otherwise, switch to state 0

```

Enter rules for Picobot

Be sure to hit "Enter rules" after making changes.

### Messages

OK

Go Stop Step Reset <-- MAP -->

0

xxxx

528

State

Surroundings

Cells to go

Previous Rule

Next Rule

West

East

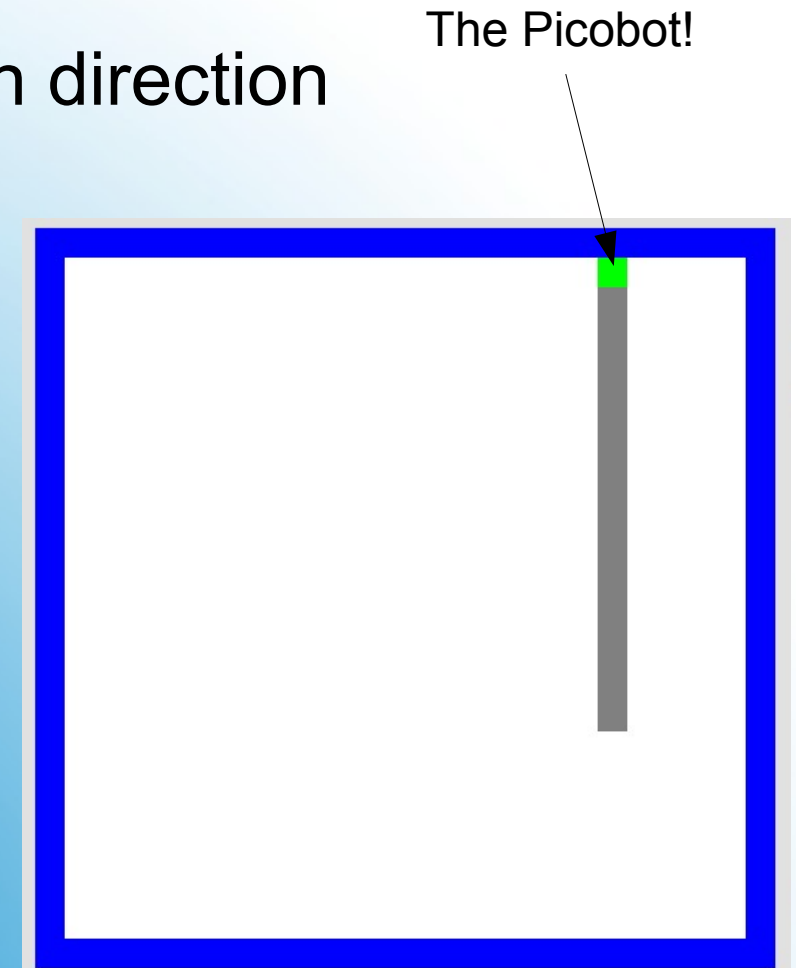
- Teleport Robot -

North

South

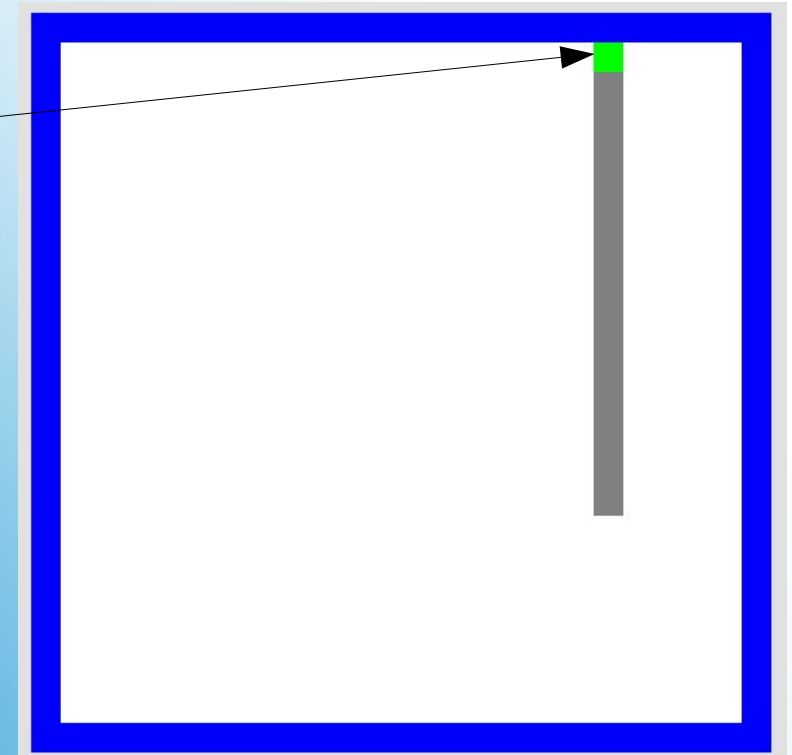
# An Introduction to Picobot

- Goal: To cover the entire environment
  - The picobot moves in each direction
  - It leaves a gray trail
  - It is guided by your instruction
- Does this remind you of anything?



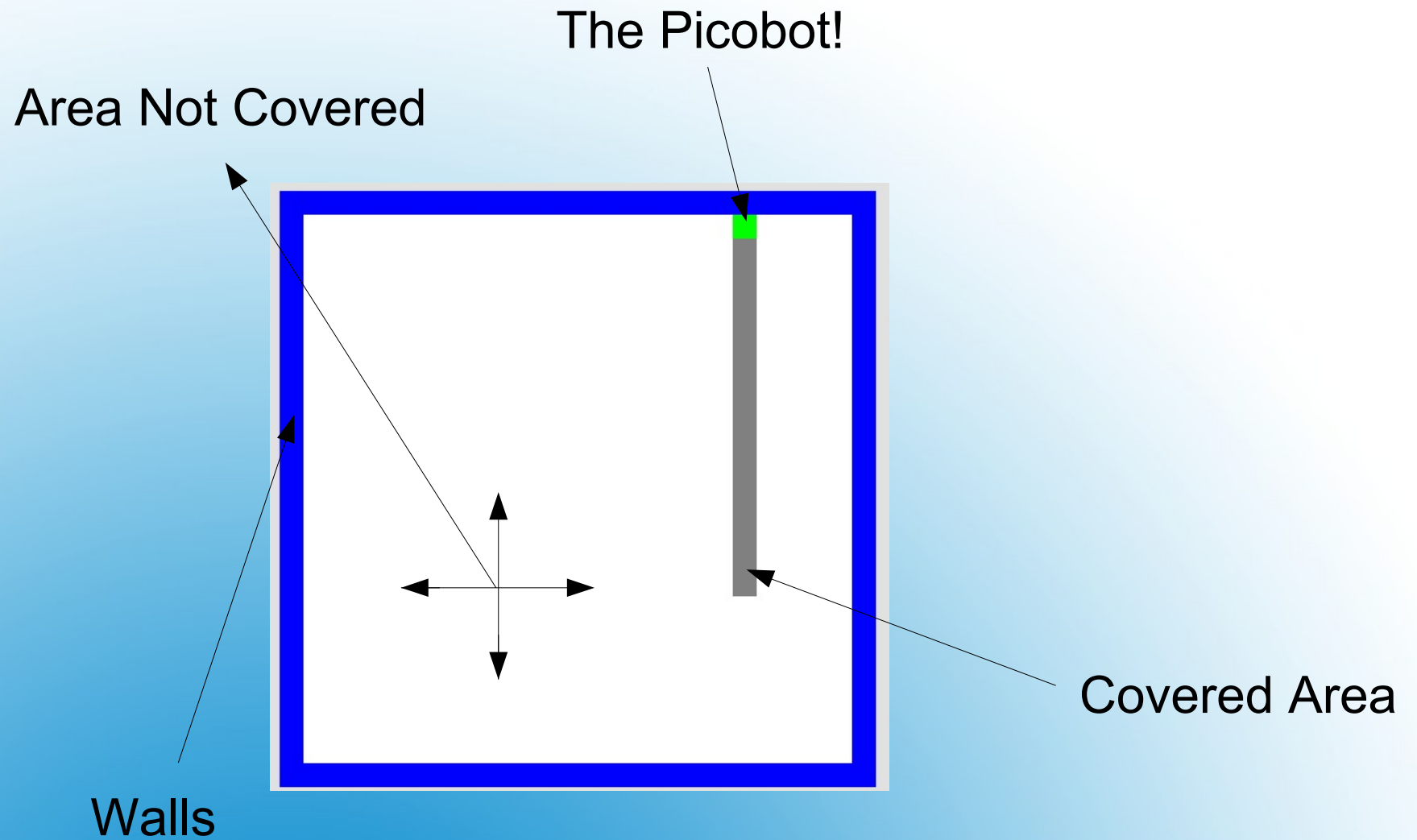
# Overview

- This represents something like a vacuum robot!

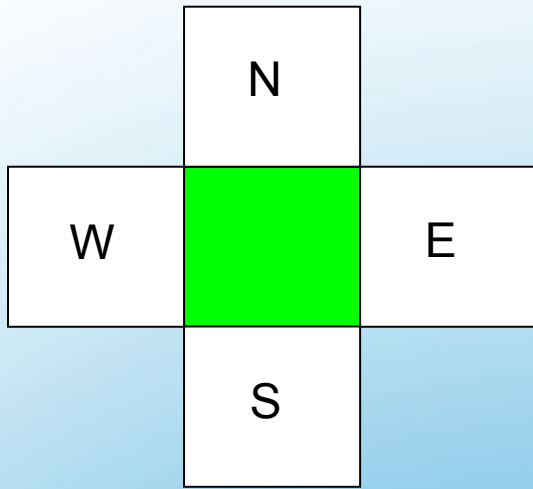


- Cannot tell a “vacuumed” area from “unvacuumed”
- May vacuum an area more than once

# Roomba!



# Environment



- Picobot can sense things in each cardinal direction:
  - North, East, West, South

- In this case, he senses his surroundings as:

**NxWx**

- They are always represented in NEWS order!

**NEWS**

# What can Picobot sense?

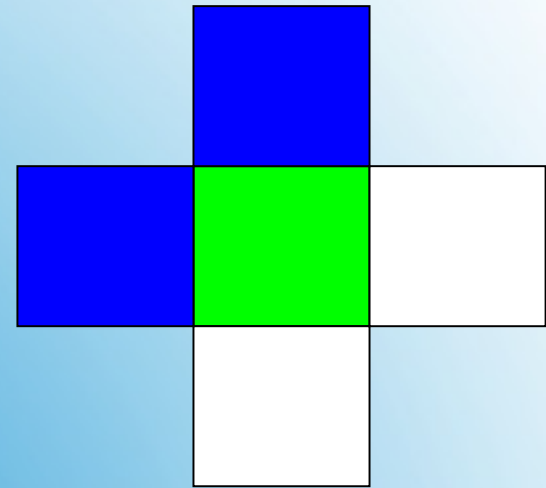
# NxWx (NEWS)

N – wall to the north

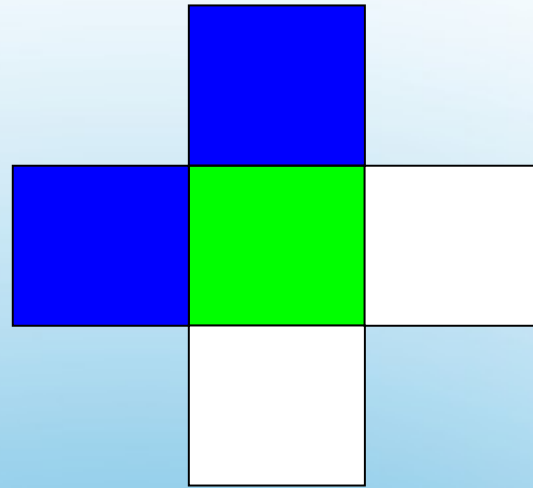
x – empty to the east

W – wall to the west

x – empty to the south

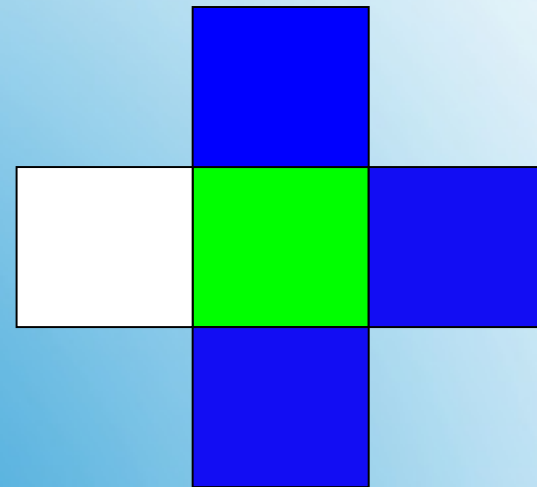
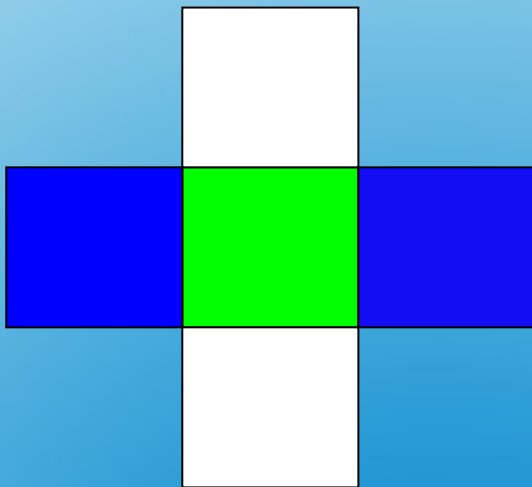


## What can Picobot sense?



N E W S

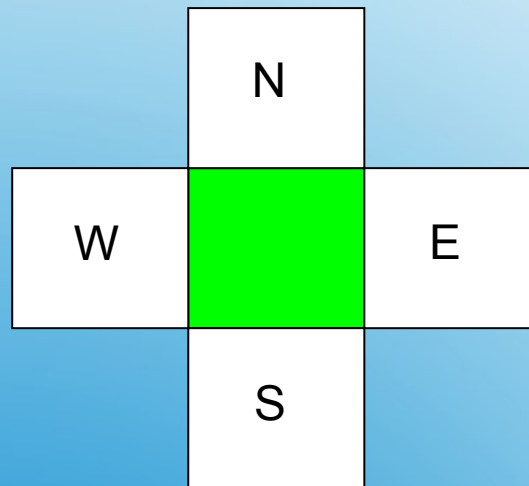
**NxWx**



What are these surroundings?

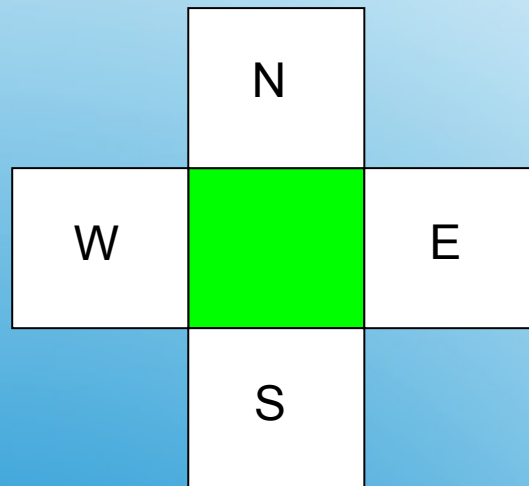


- How many unique surroundings exist?



**Question!**

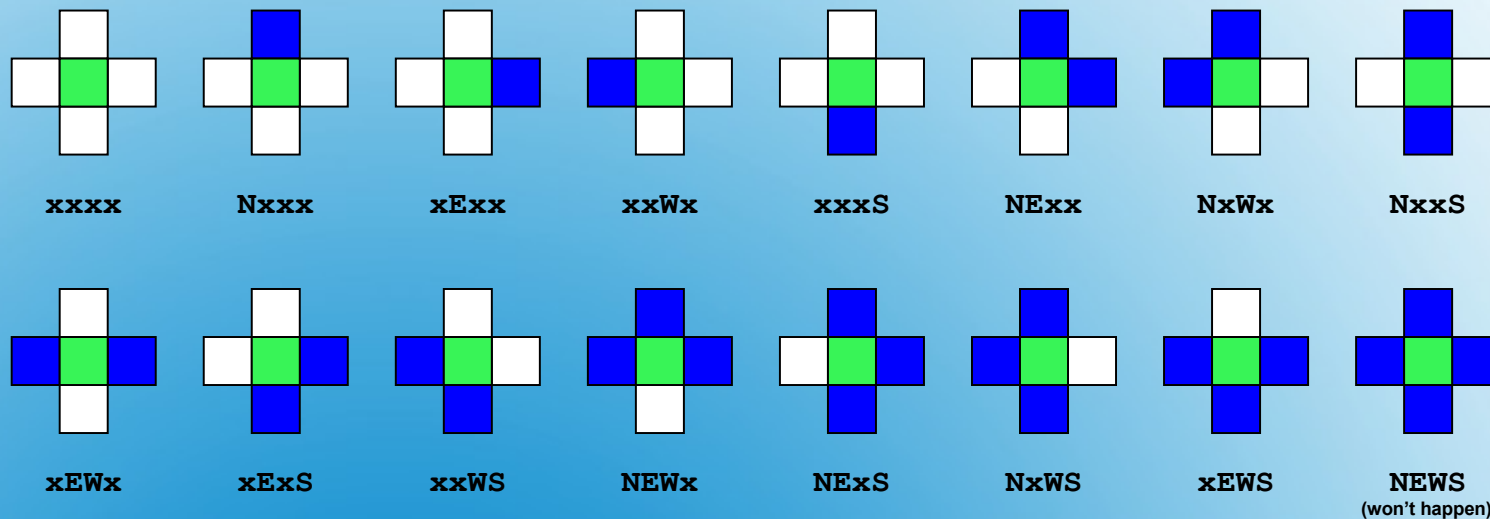
- How many unique surroundings exist?
  - Remember that for each direction, there are only two possible states (Wall or Empty)



**Hint!**

- Each direction has 2 different possibilities...

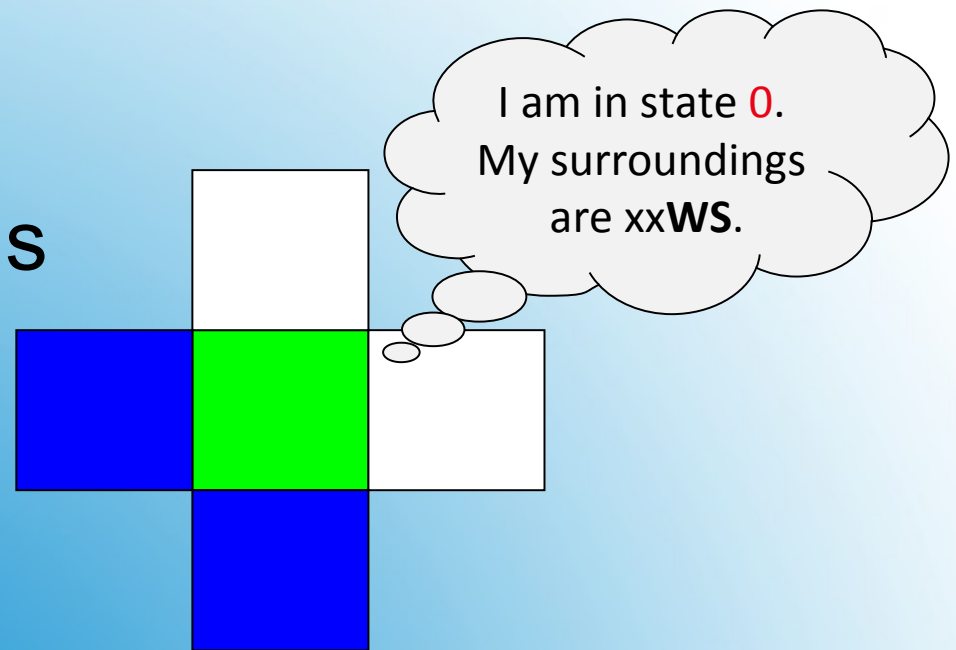
$2 * 2 * 2 * 2 == 2^4 == 16$  possible ...



# 16 Unique Surroundings

- Picobot's memory consists of a single number
  - This is called a **state**.
  - Picobot always starts at state 0.

- State and surroundings represent everything Picobot knows about the world!



## Picobot's State of Mind

state surroundings direction new state

0                    xxWS                    ->                    N                    0

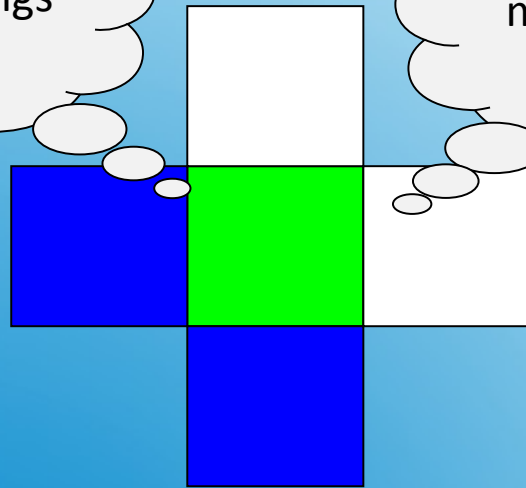
If I am in state 0, and I'm  
sensing xxWS...

Then, move north and  
“change” my state to state 0

I am in state 0.  
My surroundings  
are xxWS.

Ah! I should  
move North and  
go to state 0!

Specifying an X  
for a direction would  
cause Picobot to not  
move!



# Picobot's Instructions

state surroundings direction new state

0                    x\*\*\*                    ->                    N                    0

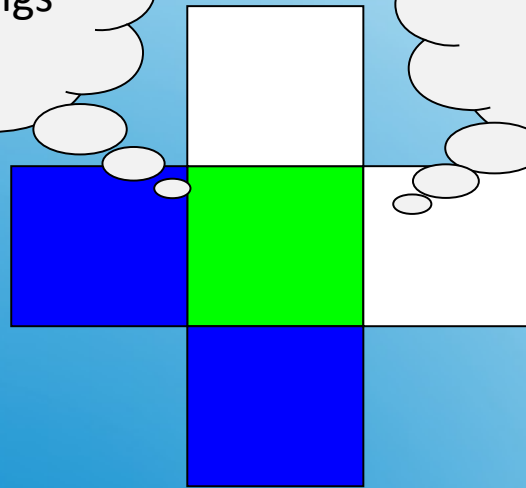
If I am in state 0, and I'm sensing  
an empty space to the North

Then, move north and  
“change” my state to state 0

I am in state 0.  
My surroundings  
are xx**WS**.

Ah! This matches  
x\*\*\*!

Asterisks (\*)  
represent both a  
wall and an  
empty space!



# Wildcard!

- Picobot reads through the list of instructions from top to bottom.
- Whenever it finds a matching rule, it executes it.
- What would these rules do?
  - What would it mean for poor Picobot?

state	surroundings	direction	new	state
0	X***	->	N	0
0	N***	->	X	0

## How Picobot chooses what to do

- Picobot reads through the list of instructions from top to bottom.
- Whenever it finds a matching rule, it executes it.

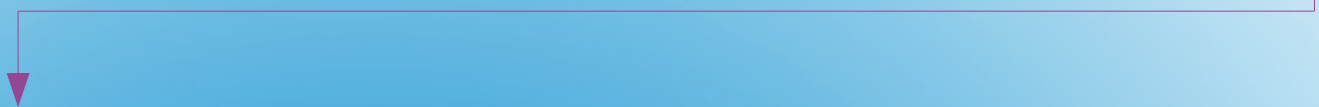
state    surroundings    direction    new    state

0            x\*\*\*            ->    N            0

0            N\*\*\*            ->    X            1

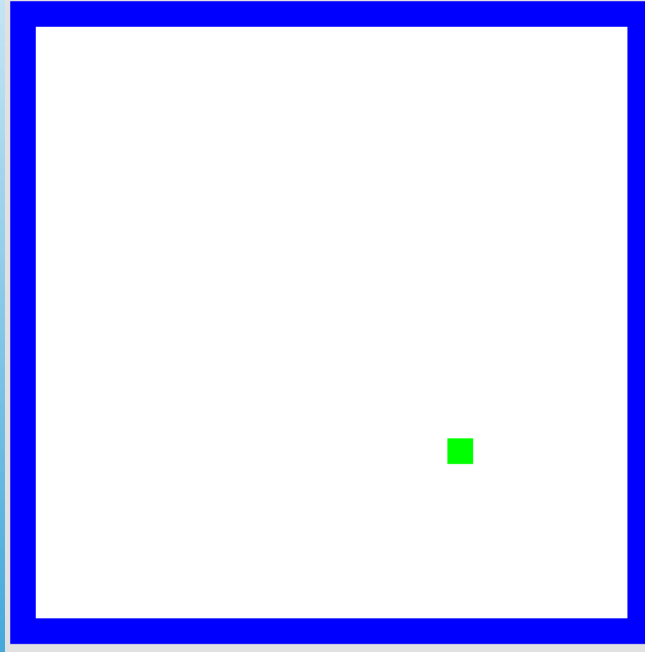
1            \*\*\*x            ->    S            1

1            \*\*\*S            ->    X            0



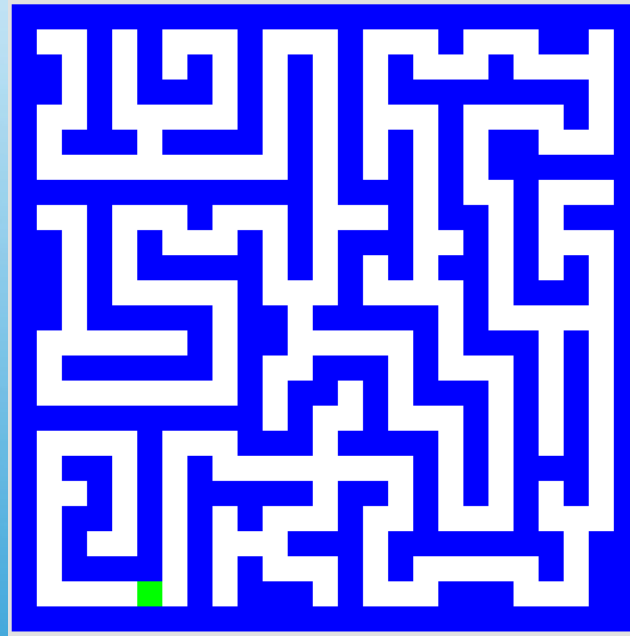
# What would Picobot do?





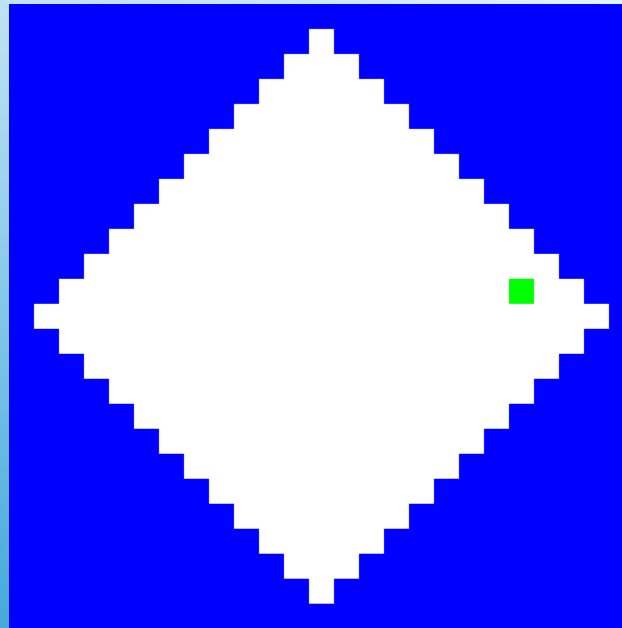
Do so regardless of starting position!

# Goal: Cover everything!



Do so regardless of starting position!

# Homework: Cover everything!



Do so regardless of starting position!

# Homework: Cover everything!

- You can look at a problem and come up with many different solutions in Picobot.
  - This is true of programming in general.
- We can measure the efficiency of our solutions!
  - The number of states and rules you have given.
  - The amount of steps your bot takes.
- *Measuring, competing, and proving that no better can be done* are all aspects of computer science and programming.

## How does Picobot measure up?