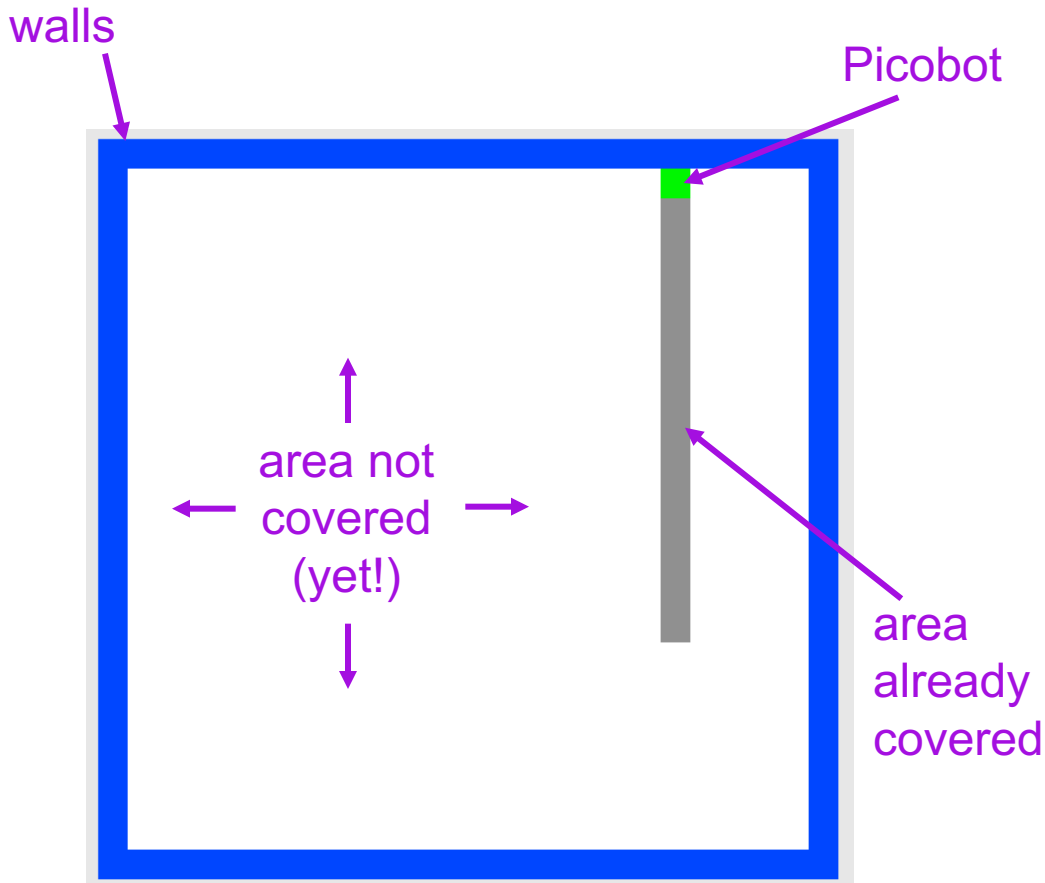
Picobot…

iRobot's Roomba vacuum

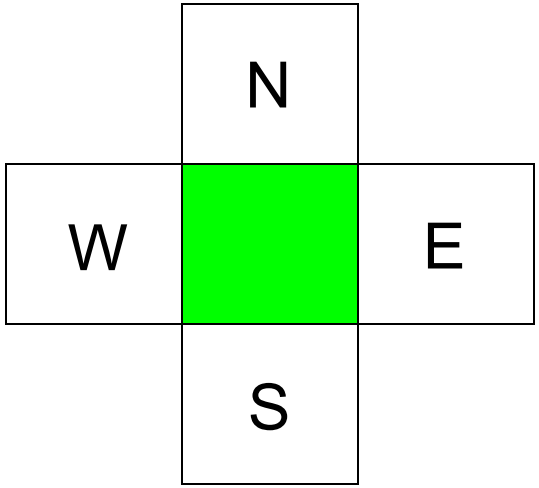
inspiration!



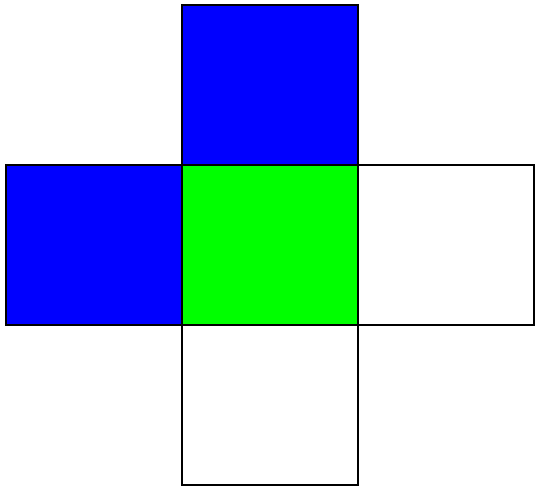
Goal: whole-environment coverage with only ***local sensing***…

Source: Harvey Mudd Computer Science Department

Surroundings



Picobot can only sense things directly to the N, E, W, and S



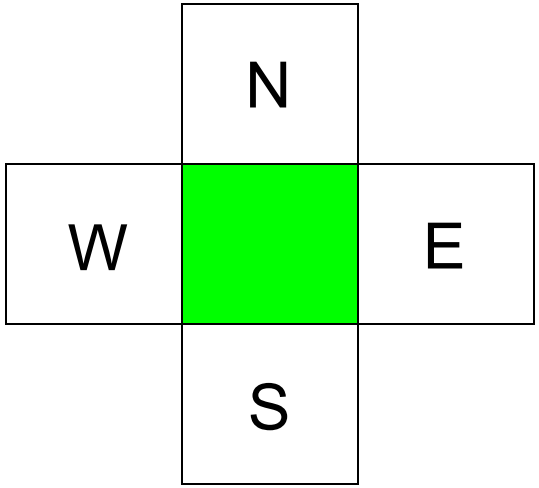
For example, here its surroundings are

Nx**W**x

N E W S

Surroundings are always in NEWS order.

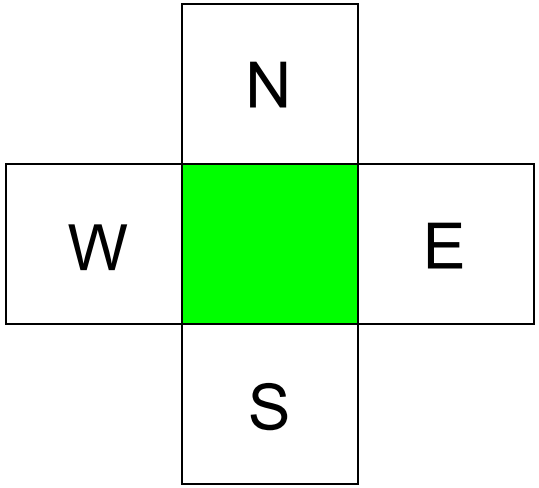
Surroundings



How many distinct surroundings are there?

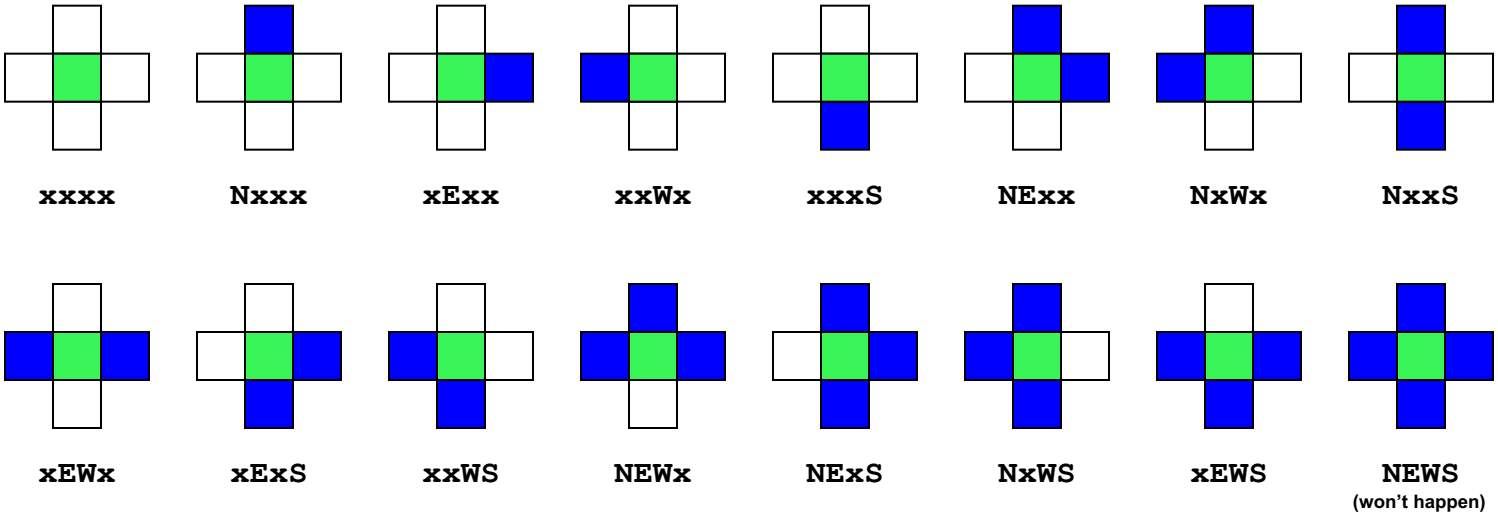
A) 6 B) 12 C) 24 D) 36

Surroundings

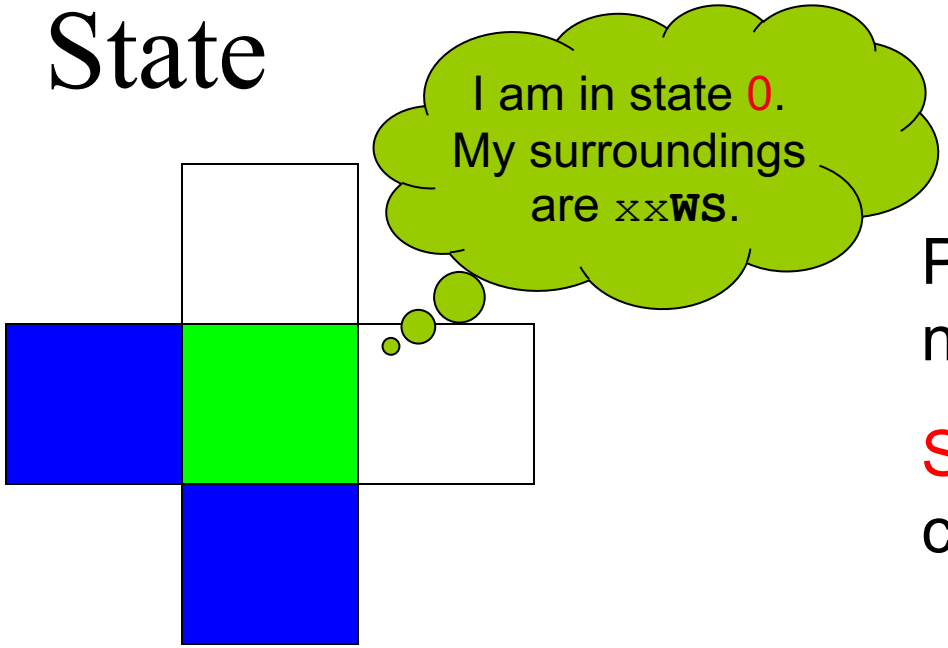


How many distinct surroundings are there?

2^4 == 16 possible …



State



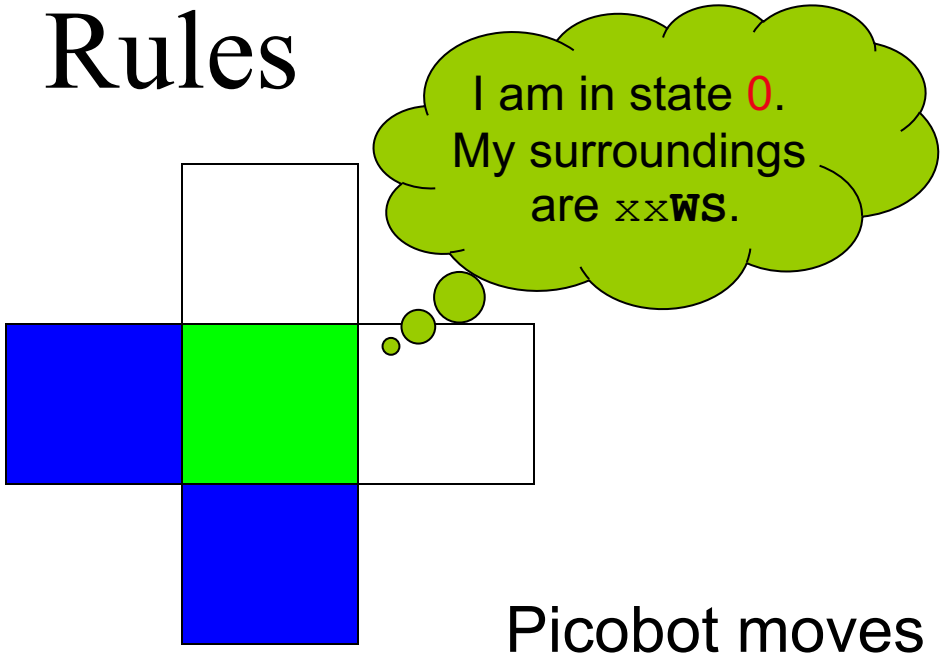
Picobot's memory is a single number, called its state.

State is the *internal context* of computation.

Picobot always <u>starts</u> in state 0.

State and **`surroundings`** represent everything the robot knows about the world

Rules



Aha!
I should move N.
I should enter state 0.

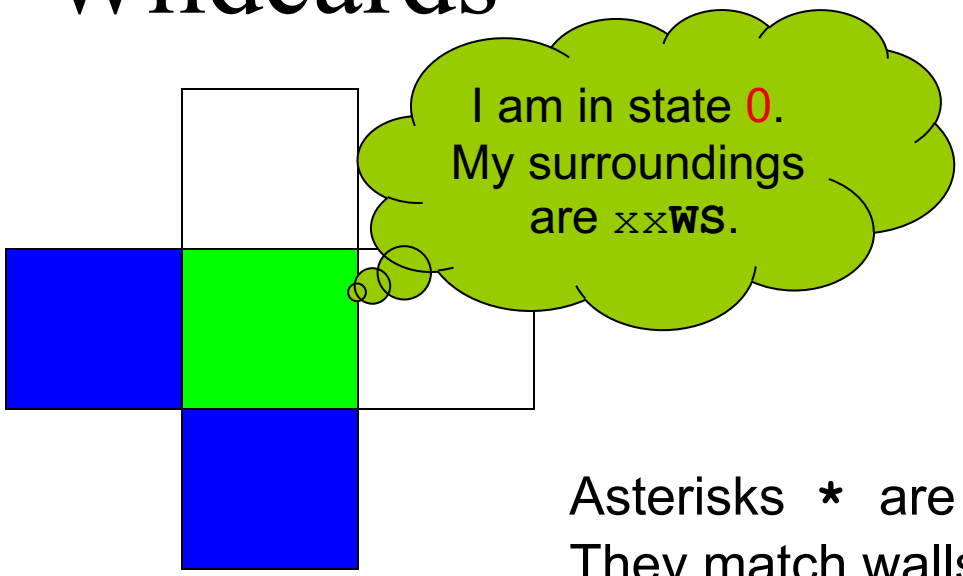
Picobot moves according to a set of rules:

state	surroundings		direction	new state
0	xx**WS**	→	N	0

If I'm in state 0
seeing xx**WS**,

Then I move **N***orth, and*
change to state 0.

Wildcards



Aha! This matches x***

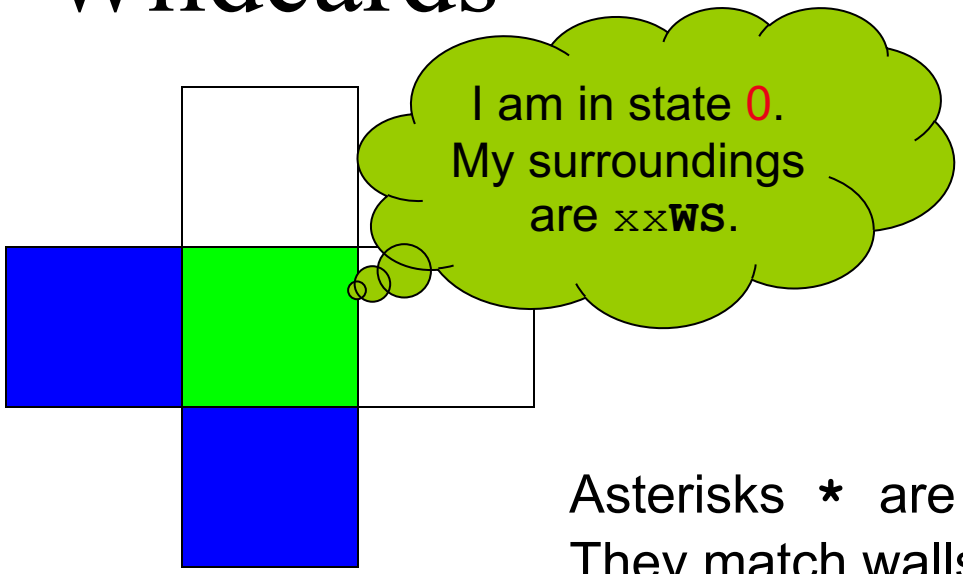
Asterisks * are wild cards.
They match walls ***or*** empty space:

state	surroundings		direction	new state
0	x***	→	N	0

and EWS may be wall <u>or</u> ***empty space***

N must be empty

Wildcards



Aha! This matches x***

Asterisks * are wild cards.
They match walls ***or*** empty space:

state	surroundings		direction	new state
0	x***	→	X	0

↑

Don't move!

Give Picobot a Set of Rules:

state	surroundings		direction	new state
0	`x***`	->	N	0
0	`N***`	->	S	0

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

What will this set of rules do to Picobot?

state	surroundings		direction	new state
0	`x***`	->	N	0
0	`N***`	->	S	0

A). Picobot will go North until it hits a wall and then it will go south.

B). Picobot will go south until it hits a wall and then it will go north

C). Picobot will go north until it hits a wall and then it will stop

D). Picobot will go north until it hits a wall, and then it will move south, then it will go north, then south, then north, never stopping.

What will this set of rules do to Picobot?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1

how can we get back down the screen?

1	*** x	→	S	1
1	*** S	→	X	0

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

What will this set of rules do to Picobot?

state	surroundings		direction	new state
0	**x***	->	N	0
0	**N***	->	X	**1**
1	***x	->	S	1
1	***S	->	X	0

Picobot checks its rules from the top each time.

When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

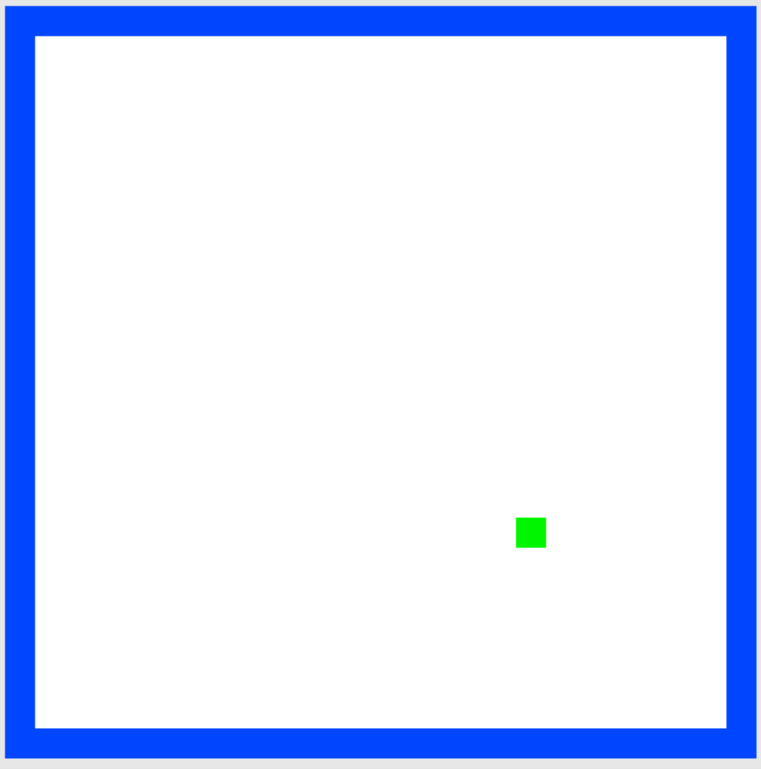
To do

Write rules that will always cover these two rooms.

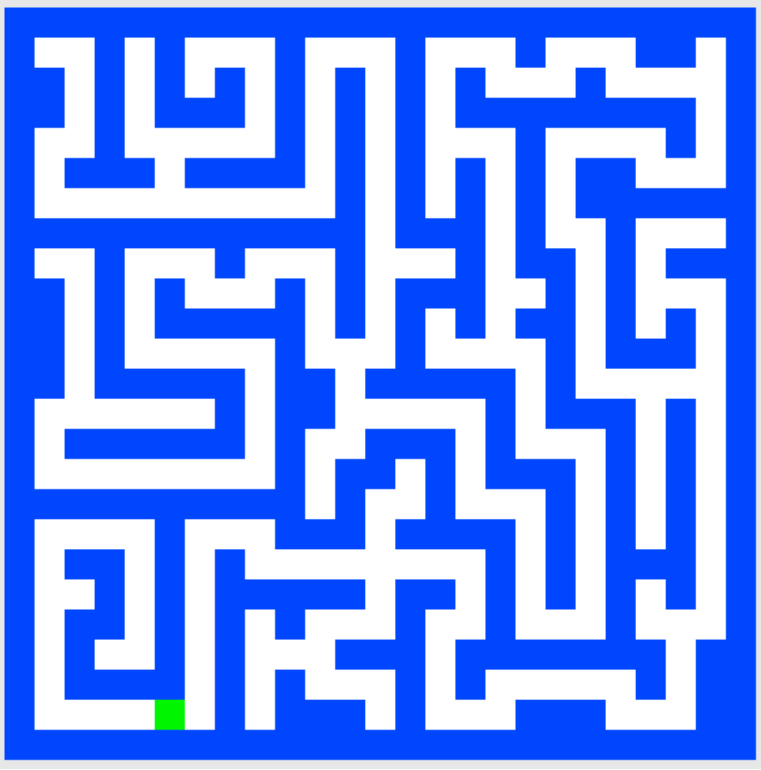
(*separate* sets of rules are encouraged…)

Picobot

Challenge #1



Challenge #2



but your rules should work *regardless* of Picobot's starting location

	f		g	
	Input		***Input***	
State	**0**	**1**	**0**	**1**
s_0	s_1	s_0	0	1
s_1	s_0	s_2	0	1
s_2	s_1	s_1	0	0

Create a finite state machine that given an input bit string $x_1x_2x_3x_4$ ... produces the output bit string $0x_1x_2x_3x_4$ ...

This is called a delay machine.

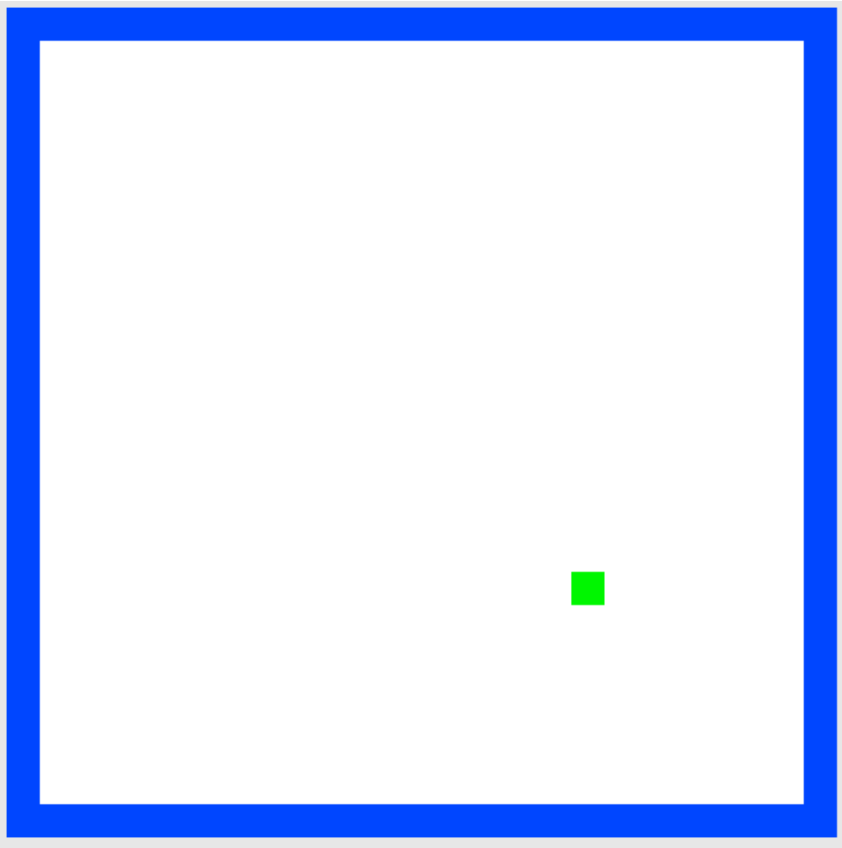
Hint: you need three states

Create a finite state machine that outputs 1 if and only if the most recent 3 input bits are 0.

Alter these "up & down" rules so that Picobot will traverse the empty room…

"Quiz"

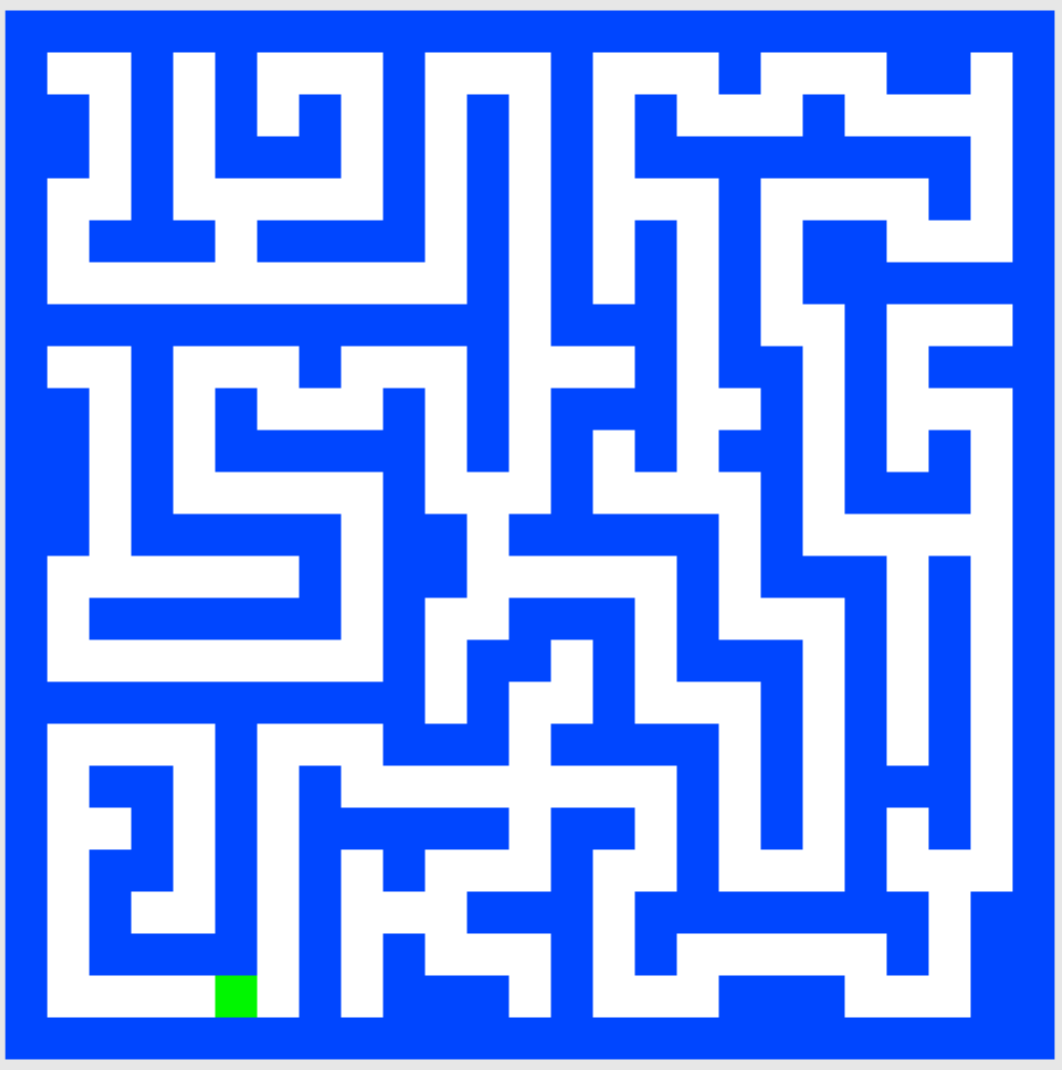
state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	X	1
1	***x	->	S	1
1	***S	->	X	0



the empty room

Hints: add E or W somewhere…
watch out for dead ends!

Ideas for the maze?



the maze

Hint: use the "right-hand-rule" !

Computer *Science*

Information is intrinsic to every system…
How can we *benefit from* this information?
"create with"

Representing it … Applying it … Measuring it

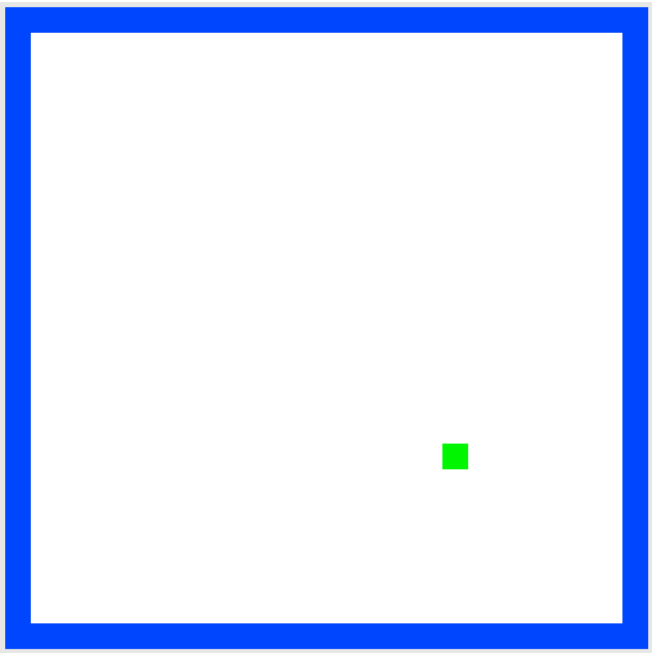
Efficiently? Effectively? Possibly?

Computer *Science*

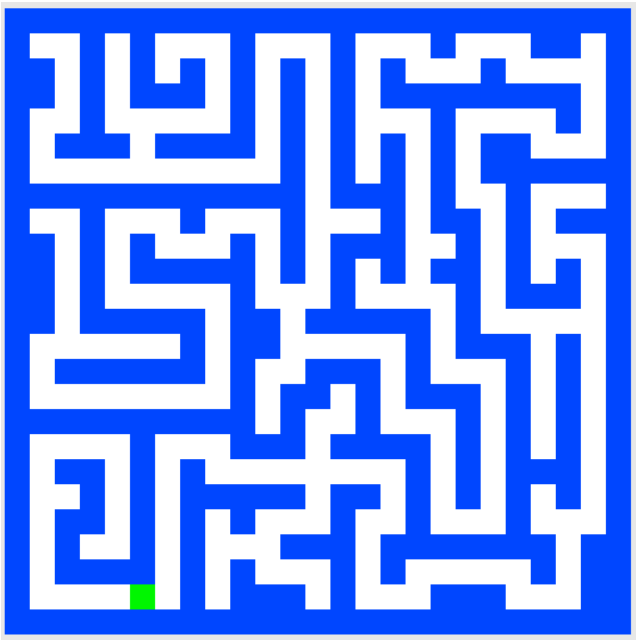
Information is intrinsic to every system…
How can we *benefit from* this information?
"create with"

Representing it … Applying it … Measuring it

Efficiently? Effectively? Possibly?



How to ***measure*** these rooms' complexity?

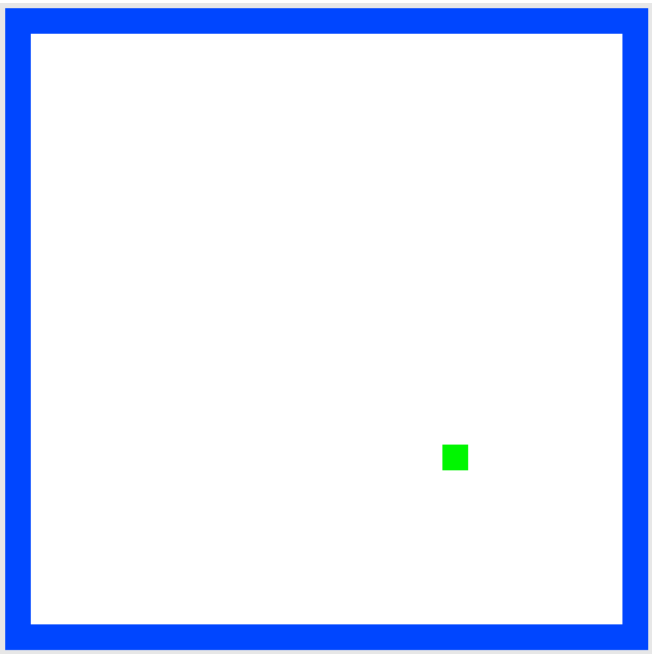


Computer *Science*

Information is intrinsic to every system…
How can we *benefit from* this information?
"create with"

Representing it … Applying it … Measuring it

Efficiently? Effectively? Possibly?

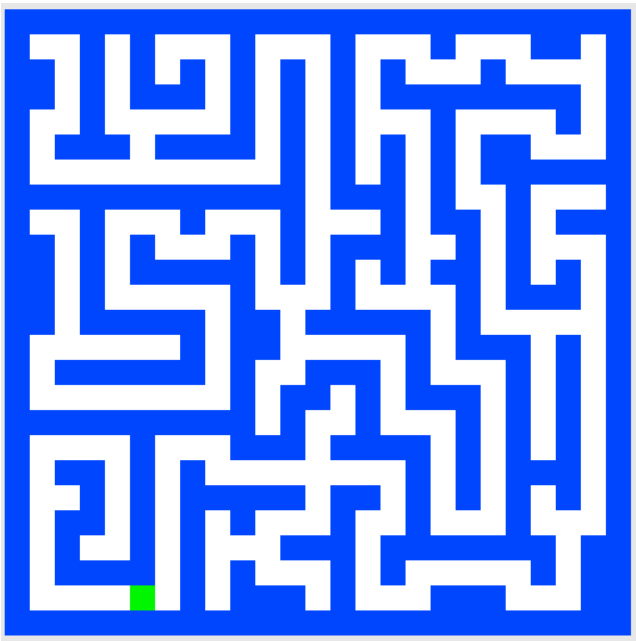


our best: 3 states, 6 rules

How to ***measure*** these rooms' complexity?

How many states and rules are really necessary ?

How much information does each system contain ?



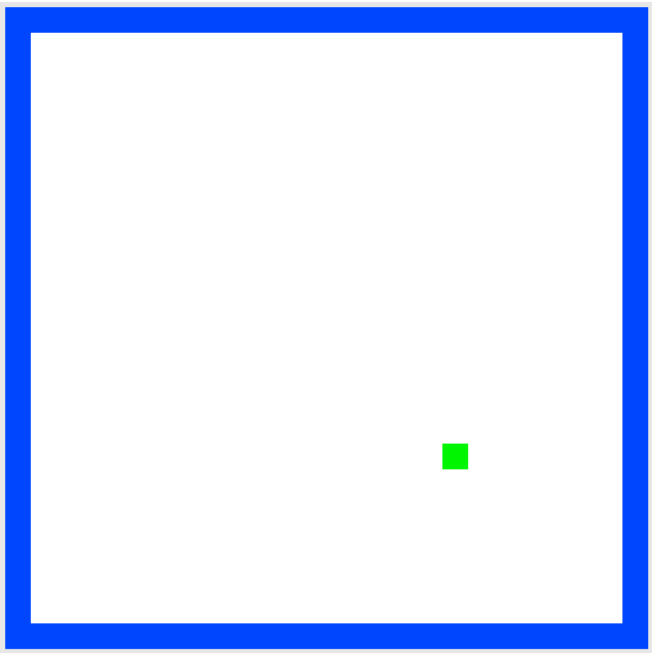
our best: 4 states, 8 rules

Computer *Science*

Information is intrinsic to every system…
How can we *benefit from* this information?
"create with"

Representing it … Applying it … Measuring it

Efficiently? Effectively? Possibly?

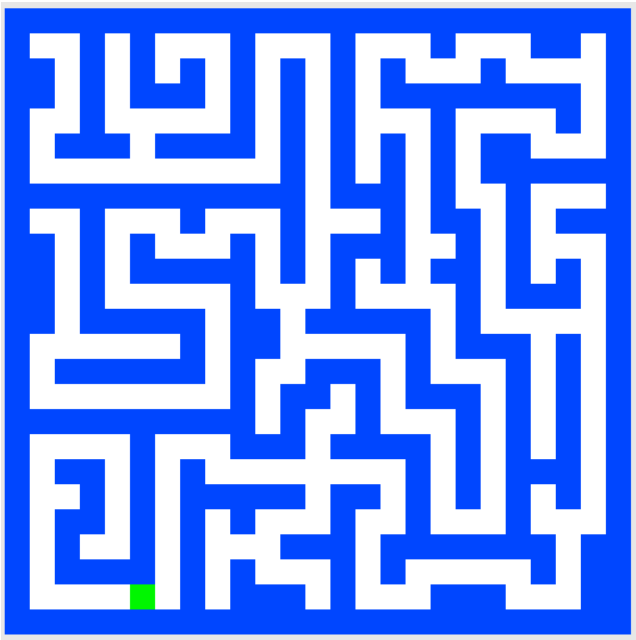


This image: 5 kilobytes

How to ***measure*** these rooms' complexity?

How many states and rules are really necessary ?

How much information does each system contain ?



This image: 20 kilobytes!

Happy Picobotting!