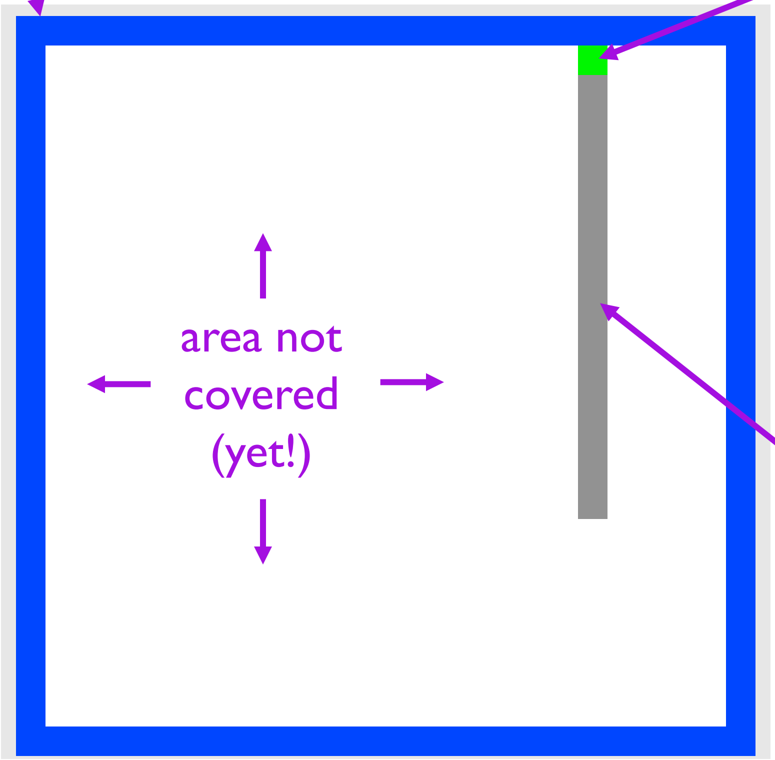
Introducing Picobot



walls

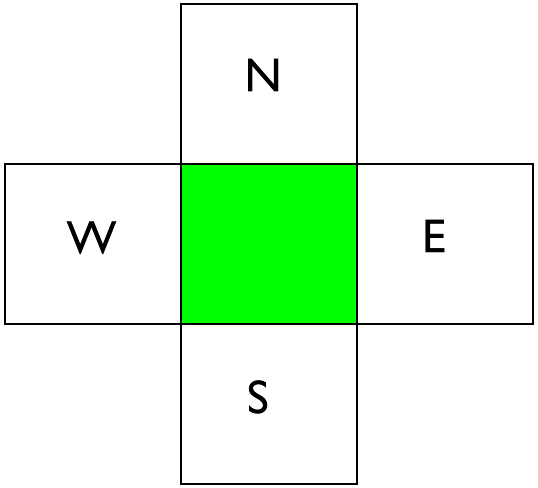
Picobot

area not covered (yet!)

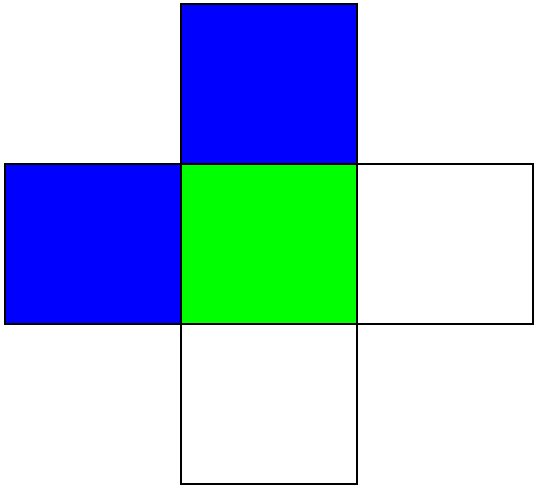
area already covered

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

Surroundings



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



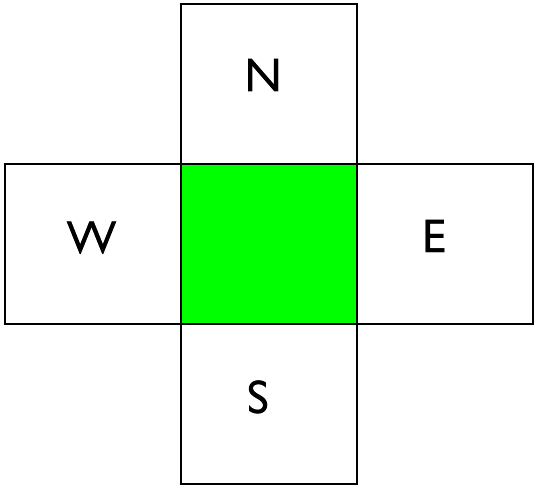
For example, here its surroundings are

NxWx

N E W S

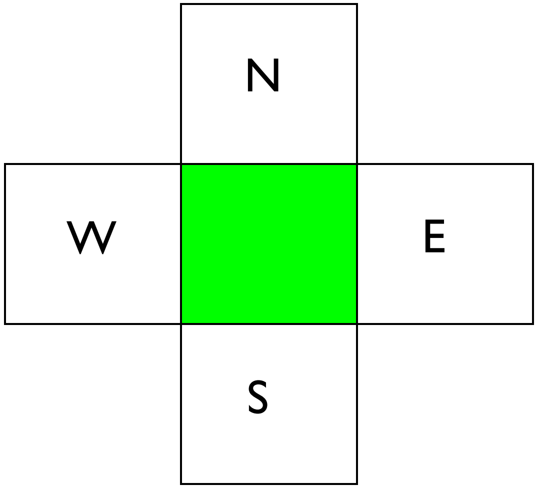
Surroundings are always in NEWS order.

Surroundings



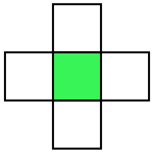
How many distinct surroundings are there?

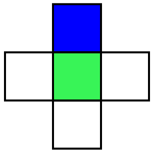
Surroundings

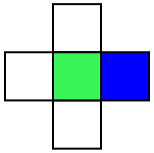


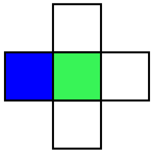
How many distinct surroundings are there?

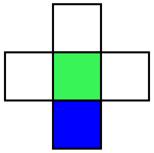
2^4 == 16 possible …

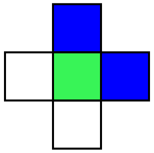

xxxx

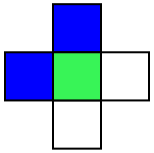

Nxxx

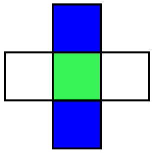

xExx

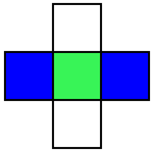

xxWx

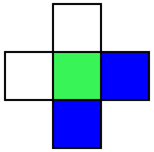

xxxS

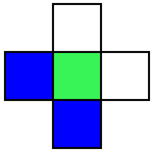

NExx

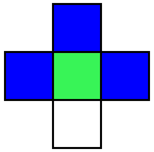

NxWx

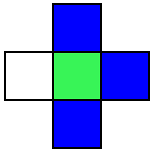

NxxS

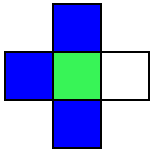

xEWx

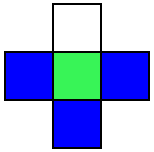

xExS

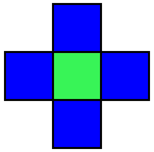

xxWS


NEWx

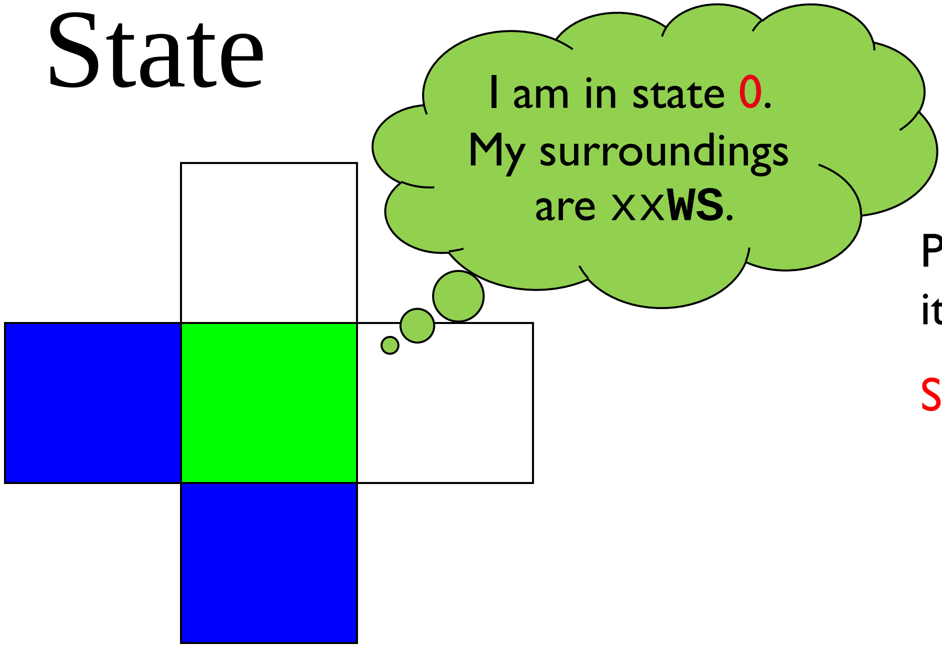

NExS


NxWS


xEWS


NEWS
(won't happen)

State



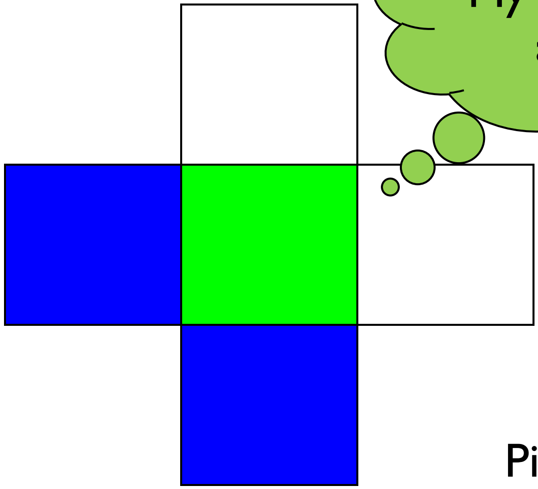
Picobot's memory is a single number, called its state.

State is the *internal context* of computation.

Picobot always starts in state 0.

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

Rules



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

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

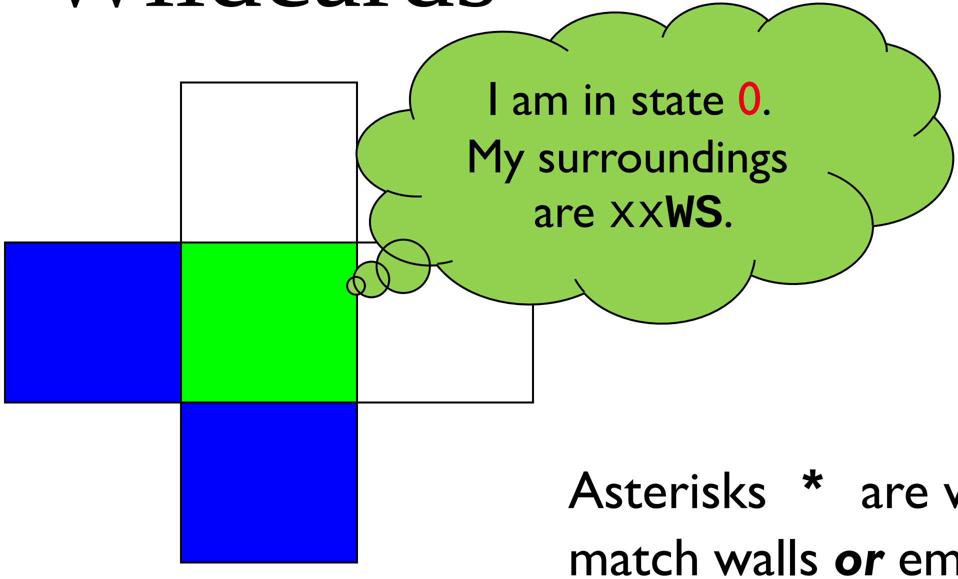
Picobot moves according to a set of rules:

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 or ***empty space***

N must be empty

What will this set of rules do to Picobot?

state	surroundings		direction	new state
0	x***	->	N	0
0	N***	->	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	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.

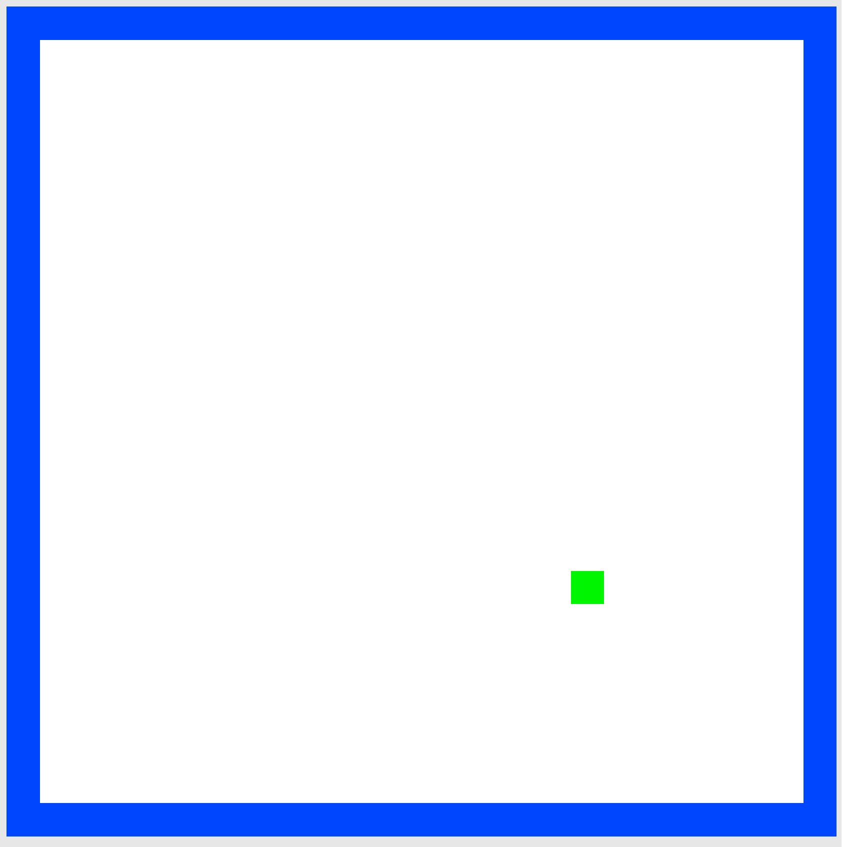
When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

Alter these "up & down" rules so that Picobot will go from side to side in the empty room instead

"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!

Getting the Job Done

- We have a specific problem to solve
  - The discrete Roomba cover-the-room problem
- We have a specific system to solve it in
  - Picobot
  - Defines the exact capabilities and limitations we can work with
- Now we can find a real solution!

- Assignment (part I due Thu, part II due Fri)