Getting started

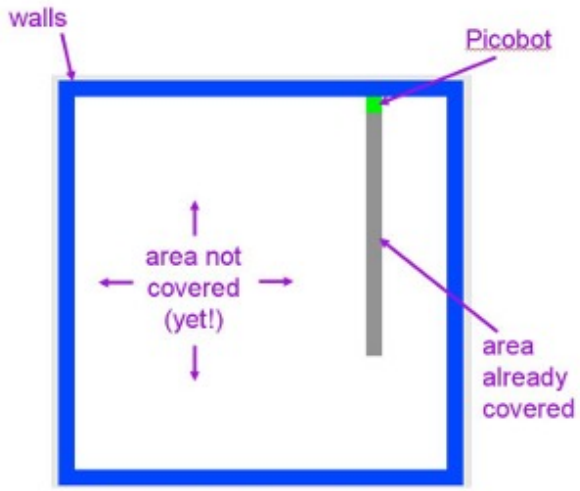
1. Course web page (syllabus, assignments, etc.): go/cs146
2. Load in-class interactor: go/cs146-inclass

FAQ:

How do I use go links? On campus you can (often) enter go links directly into the address bar and *appending a slash*, e.g., go/cs146/, and anywhere you can use go.middlebury.edu, e.g., go.middlebury.edu/cs146

What if I have COVID (or can't attend class in person)? E-mail me (before class) so I can activate a Zoom room.

Introduction to Picobot



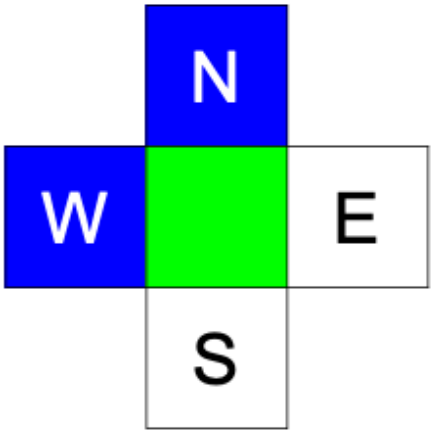
Goal: whole-environment coverage
with only *local sensing…*

https://www.cs.hmc.edu/twiki/bin/view/CSforAll/IntrotoPicobot

Picobot starts at a random location in the room and stops automatically if it has explored the entire room (the goal).

How could we explicitly define the problem?
Input: Empty room, with the Picobot at a random starting position, but always in state 0.
Output: Traverse all spaces without getting “stuck” (traversing a space multiple times is OK).
Picobot can only move in cardinal directions and only sense the presence/absence of walls immediately adjacent in a cardinal direction.

Picobot surroundings



NxWx

North East West South

Surroundings are always in NEWS order

https://www.cs.hmc.edu/twiki/bin/view/CSforAll/IntrotoPicobot

This example corresponds to a wall to the “north” and “west” sides and nothing (i.e. no wall) to the “south” and “east”.

Picobot rules


```
StateNow   Surroundings   ->   MoveDirection   NewState
```

```
0   xxxS   ->   N   0
```

"If Picobot is in state 0 and senses xxxS (only a wall to the south) it should move north and stay in state 0"

```
0   x***   ->   N   0
```

"If Picobot is in state 0 and senses nothing to the north and anything (* wildcard) to the east, west, and south (wall or not) it should move north and stay in state 0"

https://www.cs.hmc.edu/twiki/bin/view/CSforAll/IntrotoPicobot

Note that in each step the applicable rules for the current state are checked in order from top to bottom and selects the first rule that applies and then repeats that process in the next step.

One of the trickier aspects of Picobot can be the "state". State is Picobot's "memory", how it knows, for example, that it is supposed to be going "north". Recall that Picobot can only sense its local surroundings and doesn't have any overarching sense of direction (other than those local observations). The state can provide that context. More generally the state of the computer is its current condition or alternately its current internal configuration. For Picobot the state is a number from 0 - 99. That number doesn't have any intrinsic or built-in meaning. The meaning comes from you. For example, you could effectively define state 0 as the "going north till it hits a wall" state and 1 as the "going south until it can't anymore" state (or whatever numbers you choose). That is I recommended associating a goal with each state, e.g., "go all the way to the east". And when you want to switch to a new goal, you probably want to switch states.

Without using the simulator, will these rules fully traverse an empty room?

```
# Hashtag lines are optional comments

# state 0 with nothing N: go one step N
0 x*** -> N 0

# state 0 with something to the N: go W + into st 1
0 N*** -> W 1

# state 1 with nothing to the S: go one step S
1 ***x -> S 1

# state 1 with something to the S: stay put + into state 0
1 ***S -> X 0
```

A. Yes
B. No

This is our first Peer Instruction (PI) question, Each PI question will have 4 steps:

1. Solo answer: Think for yourself and select answer
2. Discuss: Analyze the problem in teams
   The goal is to practice analyzing and talking about new and challenging concepts
   Reach a single consensus answer
   If you have questions, raise your hand and I will visit
3. Group answer: Everyone in your group selects the shared consensus answer
4. Reveal and class-wide discussion
   Tell us what you talked about in your discussion that everyone should know!
   I want to hear the way your group thought about the problem. This way you are exposed to your classmates' though processes, not just mine.
   I am particularly interested in your ability to explain why the WRONG answers are WRONG.

Answer: B.
Picobot gets stuck in the upper left (northwest) corner. Picobot always starts in state 0. In most cases it will be in the middle and so traverse north until it hits a wall (triggering the second rule). At that point Picobot goes west one space and switches to state 1. With no wall to the south, Picbot will traverse south in state 1 until it hits a wall. It will then switch back to state 0 (with no movement) and start heading north. As

we noted earlier, we typically associate (define) state with a goal. In this case, state 0 could be defined as go north until a wall and then shift west one and state 1 could be defined as go south as far as it can.

Is there any starting position where these rules would successfully traverse the entire room? Yes, in the bottom right corner.

We could just test this, why do I ask you to not use the simulator? We are trying to develop a mental model of what the computer is doing. When we write our programs, we start with a mental model of what the computer should do and express that in the code. An important skill is maintaining that mental model.

The following expression made a big splash on the internet

$$8 \div 2(2+2)$$

Which of the following computes the same value assuming Python's operator precedence rules?

A. 8 // 2 * 4
B. 8 / (2 * 2 + 2)
C. 8 // (2 * 4)
D. None of the above

Answer: A

Python follows PEMDAS, or more specifically P E MD AS where MD and AS have equivalent priority, and "ties" are broken left to right. Note that 8 / 2 * 4 could also be valid depending on how strictly you interpret the above as an integer vs. floating point expression (i.e., is it 16 or 16.0). However, an even better answer is (8 // 2) * 4. Why? Because it is less confusing. While A is not ambiguous, it can be confusing and we want to write code that is easy to understand. Style matters!