

# The CS5 Times

## Eager Penguins Invade CS Course

Claremont (AP): The first-day offering of Harvey Mudd's popular CS5 course was disrupted when a large flock of penguins "Zoom-bombed" it. "They're cute," complained one distraught student, "but their squawking makes it impossible to hear the professor."

Another student disagreed. "It was easier to understand the penguins than the class material. If I can figure out where they live, I'm going to mail them some fish as thanks."

The professor eventually managed to halt the interruption by installing an aquarium screen-saver.

Office hours: see Web site  
Zoom links will *not* be posted or included in slides. See your e-mail!



Virtual penguins? ;^)

(Official course alien)

Prof. Geoff Kuenning  
<http://www.cs.hmc.edu/~geoff/geoff-schedule.html>

# Rules for Online Class

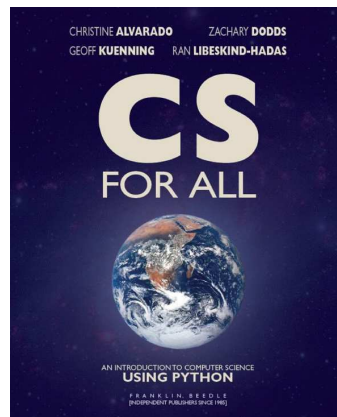
Let's admit it: this is weird

1. Use your full name (nickname preferred) on Zoom
2. Don't share or post Zoom links
3. Always have a blank sheet of paper for a worksheet
4. Keep your camera on unless your bandwidth is bad
5. Attendance (for the full session) is expected
  - E-mail me if your timezone is horrible



Can we do this inside Fortnite?

# The textbook...



Read Chapter 1!



# Overview

- Weeks 1-3: **Thinking functionally**
- Weeks 4-6: **Computer organization**
- Weeks 7-10: **Oops! (Object oriented programs)**
- Weeks 11-14: **Theoretical foundations**

**Capstone Project!**



14 weeks of action-packed excitement!

2000± languages omitted

Hello World...

C++

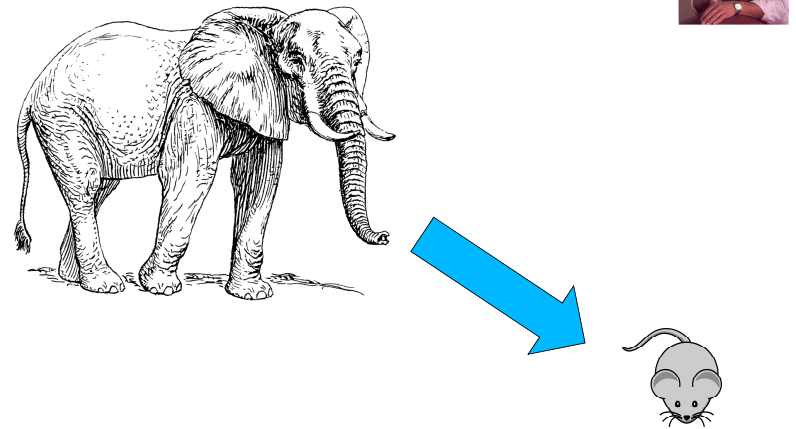
Ook

**A**TTATCG → Delete T  
**A** TATCG → Change T to C  
**A** CATCG → Insert T here  
**A** CATTC**G** → Delete G  
**A** CATTC

## Spel Cheking...



## Huffman Data Compression



## Connect 4 AI



## Picobot!

Reading: Chapter 1 in the book  
(<http://www.cs.hmc.edu/csforall/>)



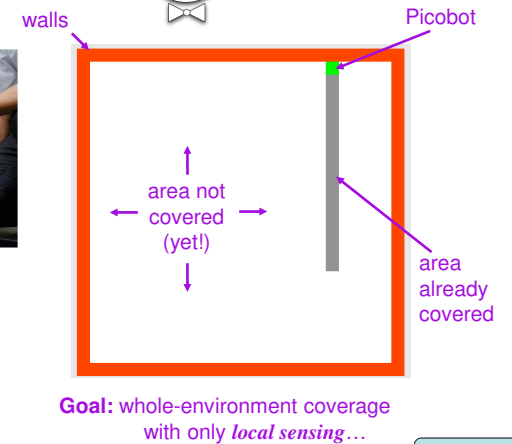
This language is not Turing-Complete. I guess that makes it "unreasonable"!



Murata Girl

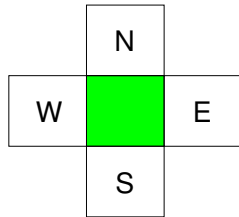


Roomba

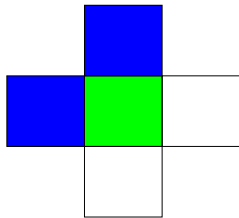


DEMO!

## Environment in the NEWS!



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



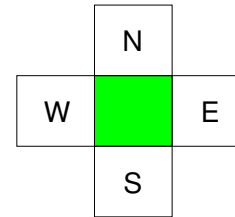
For example, here its surroundings are

**NxWx**

↑ ↑ ↑ ↑  
N E W S

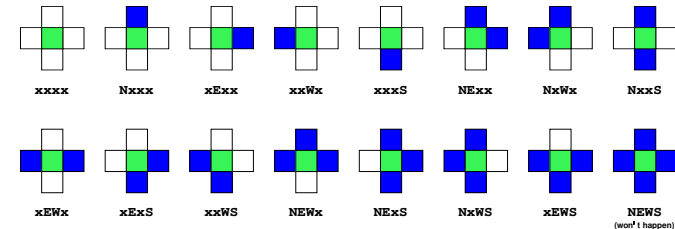
Surroundings are always in NEWS order.

## Surroundings

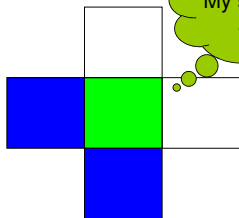


How many distinct surroundings are there?

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



## State



I am in state 0.  
My surroundings are xxWS.

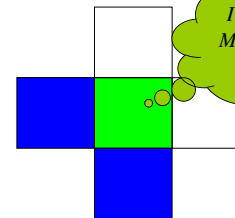
Picobot's memory is a single number, called its **state**.

**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 xxWS.

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

Picobot moves according to a set of rules:

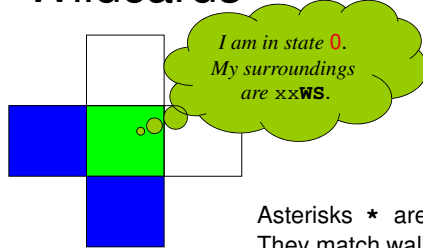
state	surroundings	→	direction	new state
0	xxWS	→	N	0

A capital "X" here means "Don't Move"

If I'm in state 0  
seeing xxWS,

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
	↑↑↑		
	<i>and EWS may be wall <u>or</u> empty space</i>		
	<i>N must be empty</i>		

What Will This Set of Rules Do to Picobot?

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

A capital "X" here means "Don't Move"

Add some code here to make Picobot go up and down in the same column forever!

Picobot checks its rules from the top each time.

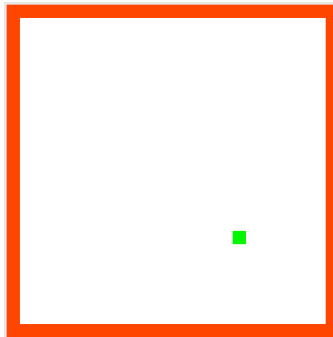
When it finds a matching rule, that rule runs.

Only one rule is allowed per state and surroundings.

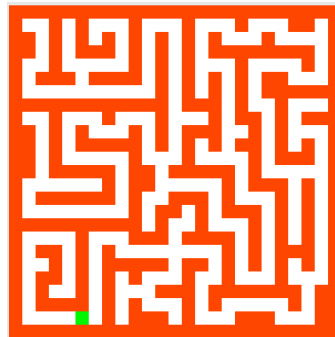
## This Week!

Write rules that will always cover these two rooms.  
(separate sets of rules are encouraged...)

### Lab Problem



### Problem 2



Your "program" can be slow but it should work for any starting location and for any wall-connected maze!

DEMO!

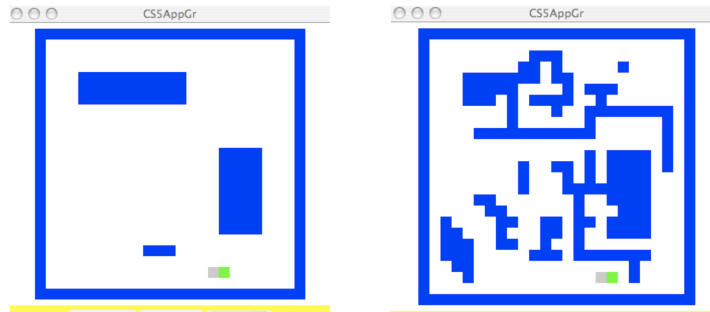
our best: 3 states, 7 rules (but Cam Zhou had 6)

our best: 4 states, 8 rules

## What's the Point?

- Simple syntax can support "powerful" computation: The picobot language syntax is very simple, yet it can control a robot in a complex environment.
- Computer scientists examine limitations of languages:
  - Are there environments that the picobot language cannot navigate?
  - If so, what features could be added to give the language more "power"?

## How About “General” Rooms?



Picobot has 100 states, but the “room” could be arbitrarily big and weird!

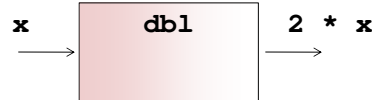
## Python and the Command Line

```
bow:2:1169> python3
Python 3.4.5 (default, Jul 03 2016, 13:32:18) [GCC] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> "Hello, world"
'Hello, world'
>>> 7*6
42
>>> import math
>>> math.pi
3.141592653589793
>>> equator = 40000 / 1.609
>>> equator / pi / 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'pi' is not defined
>>> equator / math.pi / 2
3956.6176032789394
>>> from math import pi
>>> equator / pi
7913.235206557879
>>> quit()
bow:2:1170>
```

Python makes it easy to experiment!

## Defining Your Own Functions!

```
def dbl(x):
    return 2 * x
```



```
def dbl(myArgument):
    myResult = 2 * myArgument
    return myResult
```

Notice the indentation. This is done using “tab” and it’s absolutely necessary!

VScode often indents for you!



“Outdent” with shift-tab!

## Docstrings!

```
def dbl(x):
    """This function takes a number x
    and returns 2 * x"""
    return 2 * x
```

This is sort of like teaching your programs to talk to you!



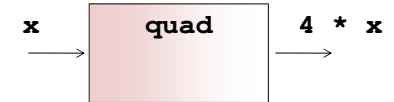
## Docstrings...and Comments

```
# Doubling program
# Author:  Ran Libeskind-Hadas
# Date:   August 27, 2011
# Time Spent: 14 hours
```

```
def dbl(x):
    """This function takes a number x
       and returns 2 * x"""
    return 2 * x
```

## Composition of Functions

```
def quad(x):
    return 4 * x
```



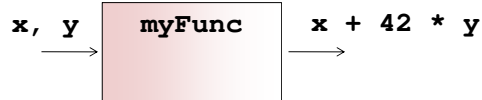
```
def quad(x):
    return dbl(dbl(x))
```



Doubly cool!



## Multiple Arguments...



```
# myFunc
# Author:  Ran Libeskind-Hadas
# Date:   August 27, 2011
```

```
def myFunc(x, y):
    """Returns x + 42 * y"""
    return x + 42 * y
```

That's a kind  
of a funky  
function!



## Mapping with Python...

```
def dbl(x):
    """returns 2 * x"""
    return 2 * x
```

```
>>> list(map(dbl, [0, 1, 2, 3, 4]))
[0, 2, 4, 6, 8]
```

```
def evens(n):
    myList = range(n)
    doubled = list(map(dbl, myList))
    return doubled
```

Alternatively....

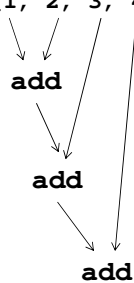
```
def evens(n):
    return list(map(dbl, range(n)))
```

## reduce-ing with Python...

```
from functools import reduce

def add(x, y):
    """returns x + y"""
    return x + y

>>> reduce(add, [1, 2, 3, 4])
10
```



## Try This...

Write a function called `span` that returns the difference between the maximum and minimum numbers in a list...

```
>>> span([3, 1, 42, 7])
41
>>> span([42, 42, 42, 42])
0
```

```
min(x, y)
max(x, y)
```

These are built in to Python!



## Google's "Secret"



This is what  
put Google on  
the map!



Research Publications

### MapReduce: Simplified Data Processing on Large Clusters

[Jeffrey Dean](#) and [Sanjay Ghemawat](#)

#### Abstract

MapReduce is a programming model and an associated implementation for processing intermediate key/value pairs, and a reduce function that merges all intermediate values paper.

Programs written in this functional style are automatically parallelized and executed on scheduling the program's execution across a set of machines, handling machine failure

## Try This...



1. Write a python function called `gauss` that accepts a positive integer `N` and returns the sum  $1 + 2 + \dots + N$
2. Write a python function called `sumOfSquares` that accepts a positive integer `N` and returns the sum  $1^2 + 2^2 + 3^2 + \dots + N^2$



You can write extra  
"helper" functions too!