Welcome to CS5 Green

Four handouts today…

Lecture notes
About you…
Worksheet
Feedback Form

“We don’t have strong enough words to describe this class” - U.S. News and Course Report

“We give this course two thumbs!” - Ebert and Roeper

“Of all the courses we’ve ever taken, this was - without a doubt - one of them!” - New York Review of Courses
Computing in the context of biological problems

How does salmonella cause disease?

Are sex determination systems in birds and mammals related?

How are Neanderthals related to modern humans?

http://foodtestinguae.blogspot.com/
http://humanorigins.si.edu/evidence/human-fossils/fossils/la-chapelle-aux-saints

Neanderthal Museum, Mettman Germany
Introductions and Course Overview

Prof. Jessica Wu

Official turtle of CS 5 Green

Guest appearances by other HMC computational biologists…
My preferred name: Jessica Wu

My preferred pronoun: She/her

My hometown: Sugar Land, Texas

Something that I really like (a food, a movie, a color, anything!)
board games, musicals

Something about me (it can be random!)
This summer, I caved for the first time.
CS 5 Green: Welcome!

Lectures, Homework Assignments, and Readings

<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday</th>
<th>Thursday</th>
<th>Homework</th>
<th>Reading in CFB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>09/03/19 - Lec 0: Introduction + picobot (J)</td>
<td>09/05/18 - Lec 1: Intro to Python (J)</td>
<td>Homework 0</td>
<td>0, 1.1-1.5</td>
</tr>
</tbody>
</table>

Next HW: Homework 0 will be due on: Tuesday, September 10 at 11:59 PM
Next Lab: Lab 0 will be held on: Thursday, September 5, 2-4 PM, Beckman 102/105
Submissions: CS submission site

Course Website
www.cs.hmc.edu/cs5green
Syllabus in a Nutshell

Lectures: Tuesday and Thursday, 9:35-10:50, Shan B442

Labs: Thursdays 2-4 PM in Beckman B105
Recommended (and incentivized), but not required

Office hours and grutoring hours on the website!

Piazza Q&A system

Homework
Lab Problem
Several additional homework problems

Pair programming encouraged on some problems
Due Tuesdays at 11:59 PM

Three CS 5 Greenbacks (aka “Euros”)
Pair Programming Policy: For some questions, you are (optionally) allowed to work as a pair. In a pair you should always program together and switch every 30 minutes.

Honor Code Policy: Other than pair programming, discussions OK, sharing or searching for code not permitted.

Grading:
- Homework + Final Project: 65%
- Midterm: (Tuesday, Oct 15, in-class): 10%
- Final Exam: (Wednesday, Dec 18, 2-5 am): 20%
- Participation/worksheets: 5% (missing up to 3 is OK)

To pass CS 5, one must have a passing grade on all components (Homework, Exams, Participation)

Pass fail vs. graded
Textbook
Q: Will I learn as much CS here as I would in CS 5 Gold?
A: Yes!

Q: Are there other courses combining CS and Bio at Mudd
A: Yes! Bio 52, MCB118b, Bio 188, and a whole major (Mathematical and Computational Biology)
Introductions: Picobot

Murata Girl

Roomba

Goal: whole-environment coverage with only local sensing...
Environment in the NEWS!

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

We can represent a particular environment with a text “code”.

Surroundings are always in NEWS order.
Surroundings

How many distinct surroundings are there?
How many distinct surroundings are there?

\[2^4 = 16 \text{ possible} \ldots\]

(xxxx) 
(Nxxx) 
(xExx) 
(xxWx) 
(xxxS) 
(NExx) 
(NxWx) 
(NxxS) 
(xEWx) 
(xExS) 
(xxWS) 
(NEWx) 
(NExS) 
(NxWS) 
(xEWS) 
(NEWS) 

(won’t happen)
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.
Picobot moves according to a set of rules:

<table>
<thead>
<tr>
<th>state</th>
<th>surroundings</th>
<th>direction</th>
<th>new state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$xx\text{WS}$</td>
<td>N</td>
<td>0</td>
</tr>
</tbody>
</table>

If I'm in state 0 seeing $xx\text{WS}$, I should move north and change to state 0.
Asterisks * are wild cards.
They match walls or empty space:

<table>
<thead>
<tr>
<th>state</th>
<th>surroundings</th>
<th>direction</th>
<th>new state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x***</td>
<td>N</td>
<td>0</td>
</tr>
</tbody>
</table>

Aha! This matches x***

I am in state 0. My surroundings are xxWS.

and EWS may be wall or empty space

N must be empty
<table>
<thead>
<tr>
<th>state</th>
<th>surroundings</th>
<th>direction</th>
<th>new state</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x***</td>
<td>-&gt;</td>
<td>N</td>
</tr>
<tr>
<td>0</td>
<td>N***</td>
<td>-&gt;</td>
<td>X</td>
</tr>
</tbody>
</table>

- 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.

1. What will this set of rules do to Picobot?

2. Try to add some rules so that we go to the bottom now and then back up forever!
1. What will this set of rules do to Picobot?

2. Try to add some rules so that we go to the bottom now and then back up forever!
This week! Write rules that will always cover these two rooms.

(separate sets of rules are encouraged…)

Lab Problem 2

Problem 4

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

our best: 3 states, 7 rules

our best: 4 states, 8 rules
Hint: a word about states

Imagine state 0 means “pointing north”
state 1 means “pointing east”
state 2 …
state 3 …
Salmonella typhimurium invading human cells

def alife_sim(num_gens, pop_size, num_to_select, network, inhibitor_l):
    """Do an artificial life simulation for numGens generations with popSize organisms."""

    # create initial population
    fit_d = {}
pop_l = []
    for org in create_initial_pop(pop_size, network, inhibitor_l):
        fitness = org.get_fitness()
        pop_l.append((fitness, org))
        fit_d[hash(org)] = fitness

    # simulate
    top_l = get_top_orgs(pop_l, num_to_select)  # get top orgs
    for i in range(num_gens):
        pop_l = []
        for j in range(pop_size):
            to_replicate = random.choice(top_l)
            new_org = to_replicate[1].replicate()  # get fitness
            if hash(new_org) in fit_d:
                fitness = fit_d[hash(new_org)]
            else:
                fitness = new_org.get_fitness()
                fit_d[hash(new_org)] = fitness
                pop_l.append((fitness, new_org))
        topL = get_top_orgs(pop_l, num_to_select)
        print("gen:", i, ":", top_l[0])
        if i%50 == 0:
            fit_d.clear()
    return top_l[0]
Why Python?

- Relatively “nice” syntax
- Emerging as language of choice in many fields
- Packages for graphics, audio, scientific computing, …

```python
print("Hello World!")
```

```java
class HelloWorld {
    static public void main( String args[] ) {
        System.out.println( "Hello World!" );
    }
}
```

```befunge
> v,v,v,v,"Hello" <
v,",","Hello World!" <
>48*, v,v,v,v,
```

Hello World...

#include <iostream.h>

main()
{
    cout << "Hello World!" << endl;
    return 0;
}
The Python interpreter
Python strings

```python
>>> biologist1 = "Watson"
>>> biologist1
'Watson'

>>> biologist2 = 'Crick'
>>> biologist2
'Crick'
```

http://blogs.nature.com/freeassociation/tag/watson-and-crick
DNA is double stranded

http://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/DNA_chemical_structure.svg/450px-DNA_chemical_structure.svg.png
Representing DNA molecules on a computer

5' - AATGCCGTGCTTGTAGACGTA - 3'
3' - TTACGGCACGAACATCTGCAT - 5'

By convention, we represent as a single string going 5' to 3'.

- Either of these two strings could be used
- These are reverse complements of each other

AATGCCGTGCTTGTAGACGTA

or

TACGTCTTCAAGCACGGCATT
Using strings: length and index

>>> myDNA = "AATGCCGTGCTT"

>>> len(myDNA)
12

>>> myDNA[0]
'A'

>>> myDNA[3]
'G'

>>> myDNA[20]
IndexError: string index out of range
Using strings: slicing

```python
>>> myDNA = "AATGCCGTGCTT"

>>> myDNA[0:4]
'AATG'

>>> myDNA[3:7]
'GCCG'

>>> myDNA[1:]
'ATGCCGTGCTT'

>>> myDNA[:4]
'AATG'

>>> myDNA[10:42]
'TT'
```
Salmonella outbreaks: routes of infection

From: http://www.cdc.gov/salmonella/outbreaks.html

Salmonella Outbreaks
2011
- Ground Turkey - Salmonella Heidelberg
- Whole, Fresh Imported Papayas - Salmonella Agona
- African Dwarf Frogs - Salmonella Typhimurium
- Alfalfa and Spicy Sprouts - Salmonella Enteritidis
- Chicks and Ducklings - Salmonella Altona
- Clinical and Teaching Microbiology Laboratories - Salmonella Typhimurium
- Turkey Burgers - Salmonella Hadar
- Cantaloupe - Salmonella Panama

2010
- Alfalfa Sprouts - Salmonella I 4,[5],12:i:-
- Shell Eggs - Salmonella Enteritidis
- Cheesy Chicken Rice Frozen Entrée - Salmonella Chester
- Frozen Marney Fruit Pulp - Salmonella Typhi (Typhoid Fever)
- Restaurant Chain A - Salmonella Hartford and Salmonella Baidon
- Frozen Rodents - Salmonella I 4,[5],12:i:-
- Alfalfa Sprouts - Salmonella Newport
- Red and Black Pepper/Italian-Style Meats - Salmonella Montevideo
- Water Frogs - Salmonella Typhimurium

2009
- Alfalfa Sprouts - Salmonella Saintpaul
- Pistachios - Salmonella (multiple types)
- Peanut Butter - Salmonella Typhimurium

2008
- Raw Produce - Salmonella Saintpaul
- Malt-O-Meal Rice/Wheat Cereals - Salmonella Agona
- Cantaloupes - Salmonella Litchfield
A more unusual route...
Sign up for ID48
Social Justice and Equity: STEM and Beyond
Wednesdays 4:15-5:30
1 unit - Shan 2440

Sign up - and try it out tomorrow!

Everyone is welcome!

You can drop it if you get busy or don’t like it!