Eager Penguins Invade Computer Lab

Claremont (AP): The first-day offering of Harvey Mudd’s popular CS5 laboratory was disrupted when a large flock of penguins took every seat in the room. “They’re cute,” complained a distraught student, “but they smell like fish and there’s no room for us.”

Professors attempted to drive the penguins away by repeatedly shouting “Shark!”, but the penguins were unmoved.

Handouts (read them all!):
– Today’s lecture notes
– A preprinted blank “worksheet”

Light reading? ;^)

Prof. Julie Medero, Olin 1269
http://www.cs.hmc.edu/~julie

(Official course alien)
Worksheet

• Your name (that you prefer to use in class)

• Your pronouns

• What do you like to do for fun?

• Something you want me to know about you
Worksheet

• Your name (that you prefer to use in class)
  Prof. Medero

• Your pronouns
  she/her/hers

• What do you like to do for fun?
  Ride bikes. Eat. Spend time with my family.
  (Eat and ride bikes with my family…)

• Something you want me to know about you
  I go to bed early!
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!
A+  
A++  
A#  
A-0 programming language  
ABAP  
ABC  
ABC ALGOL  
ABLE  
ABSET  
ABSYS  
ACC  
Accent  
ACT-III  
ATOLL - Acceptance, Test Or Launch Language  
Action!  
ACS  
ActionScript  
Actor  
Ada

2000± languages omitted

YAFIL  
Yellow - Rejected prototype for Ada  
Yorick  
Y Language  
Z notation - A program specification language, like UML  
ZOPL.  
ZPL  
ZUG  
ZZT-oop
Python

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

print("Hello World")  

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

Taking on Python…

```python
> v v   v v v, v,, "Hello" < 
> 48*, v v v, v,, "World!" < 
> 25*, @
```  

Befunge
Hello World...

```c++
#include <iostream.h>

main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```
Some Things You’ll Do This Semester…

**Sequence alignment**

**ATTATCG**
**ACATTC**

Distance is 4

**ATTAT-CG**
**A-CATTC-**
Spel Cheking…

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com

IT'S THE
BRITISH
VERSION

SPELL
CHEQUE
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”!

Goal: whole-environment coverage with only local sensing…
Stretch Break!

https://www.instagram.com/dogs_infood
Environment in the NEWS!

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

For example, here its surroundings are

\[ \text{N} \times \text{W} \times \text{X} \]

Surroundings are always in NEWS order.
Surroundings

How many distinct surroundings are there?

$2^4 = 16$ possible ...

 xxxx  Nxxx  xExx  xxWx  xxxxS  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>xs x WS</td>
<td>N</td>
<td>0</td>
</tr>
</tbody>
</table>

If I'm in state 0 seeing xs x WS, Then I move North, and change to state 0.
Asterisks * are wild cards. They match walls or empty space:

I am in state 0. My surroundings are \(xx\text{WS}..\)

Aha! This matches \(x***\)

State | Surroundings | Direction | New State
--- | --- | --- | ---
0 | \(x***\) | N | 0

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>

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.

A capital “X” here means “Don’t Move”
What Will This Set of Rules Do to Picobot?

<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>
<tr>
<td>0</td>
<td>N***</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>***x</td>
<td>S</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>***S</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

Picobot checks its rules to find one that applies
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!

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

```python
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()
```
Python and the Command Line

```python
>>> 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()
```

Python makes it easy to experiment!
Dr. Kamau Bobb!

- “Who Is A Citizen in the Modern American Technolopolis?”
- Tomorrow 7pm in SHAN 1430

Email Prof. Medero:
* A selfie at the talk
* Something you learned

…for extra credit!
Name: ________________________________________________

Date: ________________________________________________

• Your name (that you prefer to use in class)

• Your pronouns:

• What do you like to do for fun?

• Something you want me to know about you: