Introducing CS 105

Matthew Stone

Rutgers University
Department of Computer Science
and
Center for Cognitive Science

Great Insights in Computer Science

Goal: Introduce computation as a tool to think with
• You should understand what this means by the end of today’s lecture.
**Computation**

Analysis of precise, finitely-specified processes
- What can they do?
- How long do they run for?
- What do they result in?
- How do we write them down?
- When do two have the same results?

---

**Thinking**

Answering questions
Solving problems
Understanding and explaining the world
Tools for thinking

Example
Model
Metaphor
Simulation
Inference

In other words

Computer scientists think about the world by inventing and analyzing mathematical abstractions of what could happen.

So can you!
Concrete details

Useful and inspiring facts and ideas.
  • Ideas are “how to” facts.
  • One or two per lecture.

Over the course of the semester add up:
  • How computers work
  • What computer scientists do
  • How to talk to a computer scientist
  • How to think like a computer scientist

Textbook

Hillis, *Pattern on the Stone*.
  • Not too daunting or detailed, and easy to read.
  • Gives background and skeleton for lectures.
Outline of topics

Preface: Magic in the Stone
  • Why CS is about ideas
Chapter 1: Nuts and Bolts
  • Idea of a bit – a minimal distinction in state that can affect how a process evolves.
  • The physics – how bits can be built.
Chapter 2: Universal Building Blocks
  • Specifying processes – functions and machines
  • Putting them together – hardware, software

Outline of topics

Chapter 3: Programming
  • Writing down instructions how something works
  • Why computers can follow those instructions
Chapter 4: How Universal are Turing Machines
  • How many kinds of computation are there?
Chapter 5: Algorithms and Heuristics
  • Good ways to solve problems
  • Hard problems and what to do about them.
Outline of topics

Chapter 6: Memory: Information and Secret Codes
  • Bits as messages
Chapter 7: Speed: Parallel computers
  • When processes interact
Chapter 8: Computers that Learn and Adapt
  • Solving tasks that depend on new information
Chapter 9: Beyond engineering
  • Programs that evolve

Other CS 101s

Seminar in computers and society: How have computers changed the world?

Introduction to computer science: How do I learn to create my own software?

Introduction to computers and their applications: What do I need to know about computer technology?
Grading

10% Class participation
10% Homework
30% Midterm
50% Final

Class Participation

Each lecture, roughly 5 multiple choice questions
In class: 1 point for each answer, .1 bonus if right
  • Answer with clicker
On sakai: 1 point for correct answer

Thorough resources on sakai, for backup, but plan to attend lectures.
Clickers

Available at bookstores
Can keep them or sell them back
  • Most popular clicker at RU
We’ll use them for attendance, reading comprehension, quizzes, straw polls, interaction

Clicker web site: http://www.iclicker.com

Other work

Homework
  • Every other week
  • Hand in via Sakai
  • May need to run programs in “scratch”
    see http://scratch.mit.edu

Exam
  • Short answer questions modeled on HW.
Course People

Matthew Stone
http://www.cs.rutgers.edu/~mdstone
Research on language and intelligence
Office: CORE 328
Hours: Tues 3:30-5:30
Longer introduction presently.

Course People

Reid Howard
reidh@cs.rutgers.edu
Office: Hill 257
Hours: Tuesday 11-1
PhD student: theoretical CS
Course People

Ali El Qursh
elqursh@cs.rutgers.edu
Office: Core 246
Hours: Monday 1-3
PhD Student: computer vision

Course People

Rajat Roy
rajatroy@cs.rutgers.edu
Office: Hill 257
Hours: Thursday 11-1
PhD student: cognitive architecture
Illustrating computational thinking

My kindergarten classroom had a neat book
• The pages were cut like puzzle pieces
• You could turn them separately
• Offered lots of different stories…

Formal grammar

You can model the process of writing a sentence as a sequence of smaller choices.

Sentence →
    Subject
    Relation
    Object
    Modifier
Formal grammar

As long as the choices are consistent, you get a meaningful sentence as a result.

Consistency is a matter of form, not meaning.

Formal grammar

English has lots of different noun phrases
- Names: Grandma
- Definites: The firemen
- Indefinites: A cat
- Mass nouns: Dinner
- Event reference: The explosion
- Abstract nouns: Love
- Sets: several obstacles
- Quantifiers: most of the students
Formal grammar

Lots of different relationships
  Factual: got
  Negative: almost landed
  Modal: could have encountered
  Perspectival: appeared to leave

Formal grammar

Lots of different modifiers:
  • Landmarks: in the treetop
  • Containers: inside the kitchen
  • Events: at the party
  • Situations: under investigation
Formal grammar

Distinctions in meaning don’t matter for composing a sentence!

Vote

A. Grandma
B. The firemen
C. The explosion
D. Love
Vote

A. Got
B. Almost landed
C. Appeared to leave
D. Could have encountered

Vote

A. A cat
B. Dinner
C. Several obstacles
D. Most of the students
Vote

A. In the treetop
B. Inside the kitchen
C. At the party
D. Under investigation

Computational thinking

Typical features of this example:

- **Abstraction, decomposition and reduction** – Break a big task up into related tasks that are easier to carry out
- **Representation and modeling** – Formalize the distinctions that matter for task, ignoring other important differences
- **Multiple equivalent processes** – Animation of changing imagery, Sentences with varying phrases
**Computational thinking**

Concepts matter, not programming.
Depends on creativity, not routine.
Mix of theoretical and practical.
Tool for understanding things, not just making them.

**History**

Formal grammar was devised to model natural language by Chomsky (1957).

Built on theory of processes that write symbols – developed by logicians (Kleene, Post, Scott, Rabin).

Results characterize the complexity of knowledge and inference required for language use.
Key finding

Natural languages have recursion:
   Chris is writing a book that describes inventors that have built machines that changed the world that we live in.

Building sentences requires you to build other sentences inside them.
Only certain kinds of processes can do this!

Computational thinking

This is a fact about people
  • based on a computational description of what people do.
Making things is fun though

Fancier project of mine that works the same way… talking video game characters.

Our project

Develop methods for quickly building talking characters that are

expressive
interactive

That includes characters for
entertainment applications
conversational interfaces
Our proof of concept
Our proof of concept
Expressivity

The character exhibits genuine emotions and personality

Our approach:
• Draw on a human performance

Performance in our animation

<table>
<thead>
<tr>
<th>Motion</th>
<th>Voice</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>#091</td>
<td>#041</td>
<td>that was ugly</td>
</tr>
<tr>
<td>#122</td>
<td>#172</td>
<td>dude</td>
</tr>
<tr>
<td>#214</td>
<td>#174</td>
<td>you didn’t manage</td>
</tr>
<tr>
<td>#185</td>
<td>#155</td>
<td>to set up your landing</td>
</tr>
</tbody>
</table>
**Interactivity**

The character responds dynamically to ongoing events – including player actions
Interactivity

Too much to record everything

Our approach:

- Select recordings for new situations and adapt them for one another

Another selection of performance

<table>
<thead>
<tr>
<th>Motion</th>
<th>Voice</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>#031</td>
<td>#181</td>
<td>yikes</td>
</tr>
<tr>
<td>#192</td>
<td>#172</td>
<td>dude</td>
</tr>
<tr>
<td>#183</td>
<td>#133</td>
<td>on this jump</td>
</tr>
<tr>
<td>#114</td>
<td>#094</td>
<td>you got it</td>
</tr>
</tbody>
</table>
Another selection of performance

<table>
<thead>
<tr>
<th>Motion</th>
<th>Voice</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>#031</td>
<td>#181</td>
<td>yikes</td>
</tr>
<tr>
<td>#192</td>
<td>#172</td>
<td>dude</td>
</tr>
<tr>
<td>#183</td>
<td>#133</td>
<td>on this jump</td>
</tr>
<tr>
<td>#114</td>
<td>#094</td>
<td>you got it</td>
</tr>
</tbody>
</table>

Live Demo

Stone, DeCarlo, Oh, Rodriguez, Stere, Lees & Bregler
“Speaking with Hands: Creating Animated Conversational Characters from Recordings of Human Performance”,
ACM Transactions on Graphics 23(3) (SIGGRAPH 2004).
<table>
<thead>
<tr>
<th>For next time</th>
</tr>
</thead>
</table>
| Read the Preface and Chapter 1 of Hillis.  
  How does Hillis portray computational idea of **abstraction, decomposition and reduction**? |
| How does Hillis talk about **representation** – formalizing distinctions that matter? |
| How might the **universality** of computers connect to equivalence among processes? |