Engaging Students in Active Learning of Computer Science Concepts

Susan H. Rodger
Duke University

Feb. 3, 2015
ACM–iSIGCSE Workshop
Pune, India

Supported by NSF Grants DRL-1031351, DUE 1044191, CRA, and IBM Faculty Awards
Thanks for inviting me!

- Cold enough to snow back home!
- Tell you about SIGCSE from my experiences
Back in 1989

• PhD at Purdue University

• Assistant Professor at Rensselaer Polytechnic Institute

• First Task – combine first two computer science courses with automata theory
Back in 1990

• Tool – NPDA - to experiment with pushdown automata
1990 – interested in CS Education
1992 – my first SIGCSE conference
1996 – my first ITiCSE conference
2006 – my first ICER workshop
2015 – my first ACM India conference!
2015 – my first iSIGCSE event!
Never missed a SIGCSE Symposium since I started going

- 1997
- 2000

First son  Second son  Brought help
What SIGCSE has meant to me

• Great colleagues
  • Friendliest people
  • Friendliest conferences

• Learn about innovative pedagogy
  • Use in my courses

• Share
  • My ideas
  • My curriculum materials and software
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Engaging in other ways
Motivation

• Traditional way of teaching
  • Professor Lectures
  • What students hear about 87%

BLAH BLAH BLAH BLAH BLAH BLAH
Active Learning - Workshop Format
“Flipped” Classroom

• Lecture for 10-20 minutes
• Students work on problem with computers in pairs
• Bring students back together
• Pair programming
Alternative

• Work in pairs
• Everyone has their own laptop
Groups/Pairs

• Assigned

CompSci 4 Section 1
Pairs as of October 22, 2009

Front of room

G1  G2  G3  G4
G5  G6  G7  G8  G9
G10 G11 G12 G13 G14 G15
G16 G17 G18 G19 G20
G21 G22 G23 G24 G25

Group 1
Starkweather, Clara cks15@duke.edu
Dinkins, Tiffany tiffany.dinkins@duke.edu

Group 2
Listenbee, Kamerria kkl7@duke.edu
Brown, Dwayne dcb26@duke.edu

Group 3

...
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Engaging in other ways
Read the book

• Read before coming to class
  • Ready to work in class

• Reality
  • Run out of time to read, not prepared

• Bring on – Reading quizzes
  • Online
  • Turn off when class starts
Have an engaging book....

• Runescape (Brad Miller)
Electronic Textbooks (ebooks) engage students

- **OpenDSA** (Shaffer, Virgina Tech)
  - Algorithm animations built in
- **runestoneinteractive.org** (Brad Miller,)
  - Several books (Python)
    - Python - try and run code built in
    - Quizzes
- **Zyante.com** – interactive textbooks
- **Track student progress**
- **Requirements and design strategies for open source interactive computer science eBooks**
  - ITiCSE 2013 Working Group (Korhonen, Naps, et al)
Use engaging and visual tools
Python Tutor

```python
1 scores = [10, 8, 3, 9]
2 list2 = scores
3 list3 = scores[:]
4 scores[2]=5
```
Problem Solving with Feedback

- APT – Algorithmic Problem Tester
- Test one function
- Runs on multiple inputs
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Engaging in other ways
Ways to Select students to answer questions

• Problem – same students always eager

• How do you get other students to participate?
  • Randomly call on them
  • Pick A Student program
  • Work in groups – call on group
  • Assigned groups – call on group numbers
Randomly Select a Student
Pick A Student Program

Collect pictures of students

program that cycles through and randomly picks one

Remove, then start again

From Owen Astrachan
Mystery While

NETID of person 1 *
Example: abc123

NETID of person 2
Example: abc123

NETID of person 3
Example: abc123

NETID of person 4
Example: abc123

Names of people filling out form *
(first and last name for each person, separate each name by a comma)
Google Forms (cont)

What does Mystery2 do (in words)?

What does Mystery3 do? (in words)
<table>
<thead>
<tr>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What does Mystery3 do?</strong></td>
<td></td>
</tr>
<tr>
<td>It counts every character in a word except for lowercase &quot;e&quot;s.</td>
<td></td>
</tr>
<tr>
<td>It counts the number of characters in the word that aren't lowercase e's.</td>
<td></td>
</tr>
<tr>
<td>It is counting the number of characters in the word that are not e's.</td>
<td></td>
</tr>
<tr>
<td>Count the number of characters that are not 'e' in the word.</td>
<td></td>
</tr>
<tr>
<td>Mystery 3 returns a given a given word without the lowercase e's.</td>
<td></td>
</tr>
<tr>
<td>Counts the lowercase es in the word.</td>
<td></td>
</tr>
<tr>
<td>It returns the number of characters in a word that are not e.</td>
<td></td>
</tr>
<tr>
<td>Counts all of the letters in word that aren't 'e'.</td>
<td></td>
</tr>
<tr>
<td>Counts the number of characters that are not e in the word.</td>
<td></td>
</tr>
<tr>
<td>Counts all the characters that aren't e</td>
<td></td>
</tr>
<tr>
<td>counts how many letters there are that are not &quot;e&quot;</td>
<td></td>
</tr>
<tr>
<td>Mystery 3 counts the characters in a string that are not 'e', then returns the total count.</td>
<td></td>
</tr>
</tbody>
</table>
Setting up Google Forms

• Make it easy for students to get form

![Custom Bitlink](https://bitly.com/101S15-0205-01)
Outline

• Lecture formats
• Prepare for Class
• Class participation
  • Engaging with Software
    • JFLAP
    • ALICE
• Engaging in other ways
Learner Engagement Taxonomy with visualization software

• Different forms of Learner engagement
  • No Viewing
  • Viewing
  • Responding
  • Changing
  • Constructing
  • Presenting

• ITiCSE Working Group Report 2002 (Naps et al.)
Formal Languages and Automata Theory

• Traditionally taught
  • Pencil and paper exercises
  • No immediate feedback

• More mathematical than most CS courses
• Less hands-on than most CS courses
• No programming? Unlike most other CS courses
Why Develop Tools for Automata?

### Textual

\[
(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})
\]

\[
\delta = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_2), (q_2, a, q_1)\}
\]

### Tabular

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q_0</td>
<td>q_1</td>
<td>q_0</td>
</tr>
<tr>
<td>q_1</td>
<td></td>
<td>q_2</td>
</tr>
<tr>
<td>q_2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Visual

![Automaton Diagram](image1)

### Interactive

![Interactive Automaton](image2)
Overview of JFLAP

• **Java Formal Languages and Automata Package**

• Instructional tool to learn concepts of Formal Languages and Automata Theory

• Topics:
  • Regular Languages
  • Context-Free Languages
  • Recursively Enumerable Languages
  • Lsystems

• With JFLAP your creations come to life!
Thanks to Students - Worked on JFLAP and Automata Theory Tools

• NPDA - 1990, C++, Dan Caugherty

• FLAP - 1991, C++, Mark LoSacco, Greg Badros

• JFLAP - 1996-1999, Java version
  Eric Gramond, Ted Hung, Magda and Octavian Procopiuc

• Pâté, JeLLRap, Lsys
  Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer

• JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante

• JFLAP 6.0 – 2005-2008 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee

• JFLAP 7.0 - 2009 Henry Qin, Jonathan Su

• JFLAP 8.0? – 2011-14 Julian Genkins, Ian McMahon, Peggy Li, Lawrence Lin, John Godbey

Over 20 years!
DFA Example

• Build a deterministic finite automaton (DFA) to recognize even binary numbers with an even number of 1s.
• Only use symbols 0 and 1
• Binary numbers: 0, 1, 10, 11, 100, 101, 110, 111, ...
• When is a binary number an even number?
  • Ends in 0
• Which strings should be accepted?
  • 11010, 10010, 1111, 10100

No, odd
Yes
No, ends
Yes

no. of 1’s
In 1
Build with JFLAP

State q0
Start State

Transition:
- q0 to q1: 0
- q0 to q2: 1

State q1

State q2

Transition:
- q2 to q3: 1
- q2 to q4: 0

State q3

State q4

Transition:
- q4 to q3: 0

Finish State

Graph shows a finite state automaton with states q0, q1, q2, q3, and q4, and transitions labeled with 0 and 1.
Simulation on 1101010
Simulation on 1101010
Simulation on 1101010
Simulation on 1101010
Simulation on 1101010
Simulation on
1101010
Simulation on 1101010
Accepts Input!

1101010
Add meaning to states!
A deterministic finite automaton (DFA) for recognizing strings with an even number of 1's.

- **States:**
  - q0
  - q1
  - q2
  - q3
  - q4

- **Transitions:**
  - From q0:
    - On input 1, go to q2.
    - On input 0, go to q1.
  - From q1:
    - On input 0, go to q3.
    - On input 1, stay at q1.
  - From q2:
    - On input 0, stay at q2.
    - On input 1, go to q3.
  - From q3:
    - On input 0, go to q4.
    - On input 1, stay at q3.
  - From q4:
    - On input 0, go to q4.
    - On input 1, stay at q4.

**Accepting States:** q4

**Transition for an Even Number of 1's:**
- Start at q0.
- If the number of 1's is even, end at q4.
- If the number of 1's is odd, end at q3.

**States Descriptions:**
- q0: Start state
- q1: State with only one 0
- q2: Odd number of 1's
- q3: On input 1, stay
- q4: Accepting state

The DFA checks if the input string has an even number of 1's.
The diagram represents a finite state machine (FSM) used to determine the number of 1's in a binary input.

- **q0**: Start state. Transition on 1 to q2.
- **q1**: State with only one 0 input. Transition on 0 to q2.
- **q2**: State with an odd number of 1's. Transition on 0 to q4.
- **q3**: State with an even number of 1's, ends in 1. Transition on 0 to q4.

The states q2 and q3 are key in determining the parity of the number of 1's in the input.
Test Multiple Inputs

### Table of Results

<table>
<thead>
<tr>
<th>Input</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>Reject</td>
</tr>
<tr>
<td>111</td>
<td>Reject</td>
</tr>
<tr>
<td>1010</td>
<td>Accept</td>
</tr>
<tr>
<td>10110</td>
<td>Reject</td>
</tr>
<tr>
<td>101</td>
<td>Reject</td>
</tr>
<tr>
<td>1100</td>
<td>Accept</td>
</tr>
<tr>
<td>110110</td>
<td>Accept</td>
</tr>
</tbody>
</table>
Alice Programming Language

• Create interactive stories or games
• Learn programming in an easy way, drag-and-drop your code
• Problem solving with visual feedback
  • Logical thinking, Computational thinking
• Along the way, learn computer science concepts:
  • Loops, classes, methods, functions, arrays
Alice Developed by Randy Pausch

• Carnegie Mellon University
• Virtual Reality Researcher
• Wrote the Last Lecture
• Died of Pancreatic Cancer in 2008
More on ... Alice Programming Language

- Has libraries of 3D objects

- Keeps Track of objects you select
Objects Have Multiple Parts that are moveable
Object Position

• Objects
  • Are positioned in 3D space
  • Have six degrees of freedom
Alice Code is Easy to Learn

Select Code, Drag-and-Drop code in program
Play Alice Animation

- Chicken rises, cow turns head and talks
Adventures in Alice Programming
Grades 5-12 Outreach

www.cs.duke.edu/csed/alice/aliceInSchools
Adventures in Alice Programming

- 2-week Teacher workshops
  - Over 200 teachers, middle school, high school, some elementary
  - First week Teach Alice, Practice
  - Second week - Develop Lesson Plans
  - One-week follow-up workshop the following summer
  - Summers 2008-2015, funding for lodging

- Main Sites:
  - Duke University, Durham, NC
  - Charleston/Columbia, SC
  - San Jose, CA (starting 2014)
Science Example
How a volcano is formed
Science – Population Change
Science – Population Change (end)

now we'll graph the data in a bar chart to see how the population changed over time.
¡Bienvenido al programa de cocinar!
Focus on math
Math Example – Plotting Numbers

I am going on a bike ride
Math Story on Fractions
Curriculum Materials

• Over 90 tutorials available for free
• Beginner, advanced, challenges, projects
• Paper handouts and video
• Teacher lesson plans in many disciplines
Example: Getting Started Tutorial teaches:

• Placing objects
• Moving objects
• Setting up Camera tripods and moving between views
• Using built in methods and writing your own
• Gluing objects together
• Adding sound, 2D pictures to enhance world
Getting Started Tutorial – 3 part
Sample tutorial: Scene Change
New Tutorial – Camera views following a person
Tutorial for Simple Game – Control boat, earn points

0.0

Click Instructions to Start

To win this game, you must steer the boat through each ring and beat the clock. You receive one point for each ring, and there are 10 rings, so if your score is less than 10 at the end, you lose!
Fun with Alice

The ITiCSE 2014 boat trip
What a 6th grader can do with Alice - teacher Chari Distler
No Superheros in Alice
Outline

• Lecture formats
• Prepare for Class
• Class participation
• Engaging with Software
• Engaging in other ways
Example: Be a Robot

- 4 People
  - Controller (head)
  - Sensors (eyes)
  - Manipulators (2 hands)
- Blindfolded except eyes
- Controller knows what to build
- Limited communication

SIGCSE 96,
Rodger, Walker
CS Concepts Coming Alive- Back in 1989

• New Assistant Professor at Rensselaer

• What data structure is this?
Engaging students in a group activities/large course

• Problem Solving in groups
  • Clickers, Google forms – compare results
  • Flip Classroom, reading quizzes (turn off at start of class)

• Acting out stories, games
  • *Everything I needed to know about teaching*... - Pollard, Duvall (SIGCSE 2007)

• Acting out algorithms with the whole class
  • Make a binary tree with the whole class
  • Calculate the height of the tree
  • *Making Lemonade ... large lecture classes* – Wolfman (SIGCSE 2002)

• Acting out algorithms with a subset of students
  • Sorting algorithms – selection sort, insertionsort, etc
  • CS Unplugged activities
Middle School students sorting themselves with Bubblesort
Cards

- Card Class – shuffling, dealing hands
- Poker hands – Full house, Flush, etc.
Royal Flush
Notable Women in Computing Cards
bit.ly/NotableW

• Based on Wikipedia project – wrote guide on how to write a Wikipedia page on a Notable women in Computing
• Picked 54 Women - deck of cards
• Page on using cards to teach CS
• Poster of the women
3 female Turing Award Winners
6 women Eniac Programmers
Anita Borg
Grace Hopper
ACM Fellows
Other
Interaction in Class – Props

Passing “Parameters” in Class

• Pass by reference – throw frisbee

• Pass by value – throw copy of frisbee

• Pass by const reference – throw “protected” frisbee
Interaction in Class – Props
Linked List and Memory Heaps

ITiCSE 98 – Astrachan – “Concrete Teaching: Hooks and Props as Instructional Technology”
Interaction in Class – Props
Memory Heap
Paper Fetch!

If you’ve ever seen one of those wonders of nature movies where lizards catch flies on their tongues, you’ll pick up this game much more quickly.

1. Lay five or six pieces of paper on the floor, in a circle with a diameter of, say, 6 or 7 feet. (The bigger the circle, the harder the game.) Get a partner with a similar fascination with lizards.
The edible way to engage students
About Me - Hobby – Baking Shape cakes

The Wiggles magazine
Issue No. 42
How do you make those cakes?
What happens when your hobby and your career collide?

It is now time for engaging students with edible CS
CS 1
Sorting Cookies
• TM for $f(x)=2x$ where $x$ is unary

• TM is not correct, can you fix it? Then eat it!

• States are blueberry muffins
Students building DFA with cookies and icing
Alice Programming Language
Conclusions

• We have shown several ways to engage students with CS concepts and make those concepts come alive

• Questions?