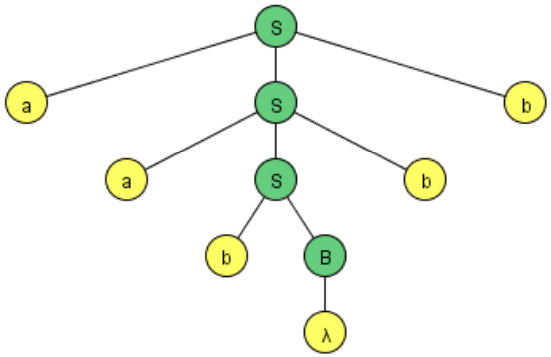
Computer Science Concepts Come Alive



Susan H. Rodger
Duke University
rodger@cs.duke.edu

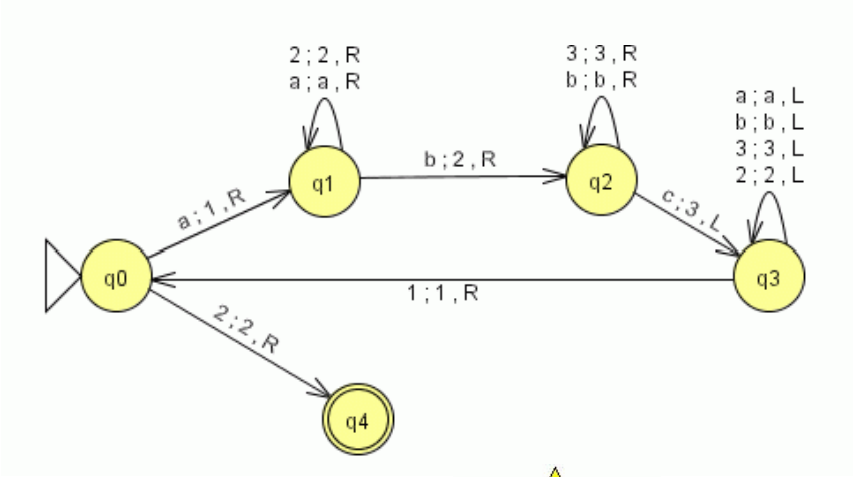
0	bear
1	elephant
2	gorilla
3	kangaroo
4	horse
5	zebra
6	tiger

Helsinki, Finland
December 15 and 16, 2009

Supported by NSF Grants DUE 0442513
and ESI 0624642, CRA, and IBM Faculty Awards

Three Projects I'm involved in

- JFLAP
  - Software for automata theory
  - Study with 14 universities



- JAWAA
  - Algorithm animation

- The Alice project
  - Create 3D virtual worlds
  - Teaching programming non-majors college
  - Teaching to K-12 (6 regional sites)

My Research Areas

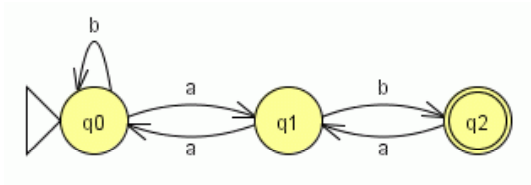
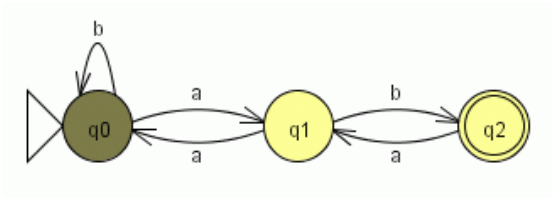
- Computer Science Education
- Visualization and Interaction
  - Instructional Tools for Theoretical concepts
    - Automata theory and formal languages
- Algorithm Animation

Motivation for Developing Visualization Software …

Formal Languages and Automata Theory

- Traditionally taught
  - Pencil and paper exercises
    - No immediate feedback
    - Limited – simple examples
- Different
  - More mathematical than most CS courses
  - Less hands-on than most CS courses
  - Programming is in most of their CS courses, not here

Why Develop Tools for Automata?

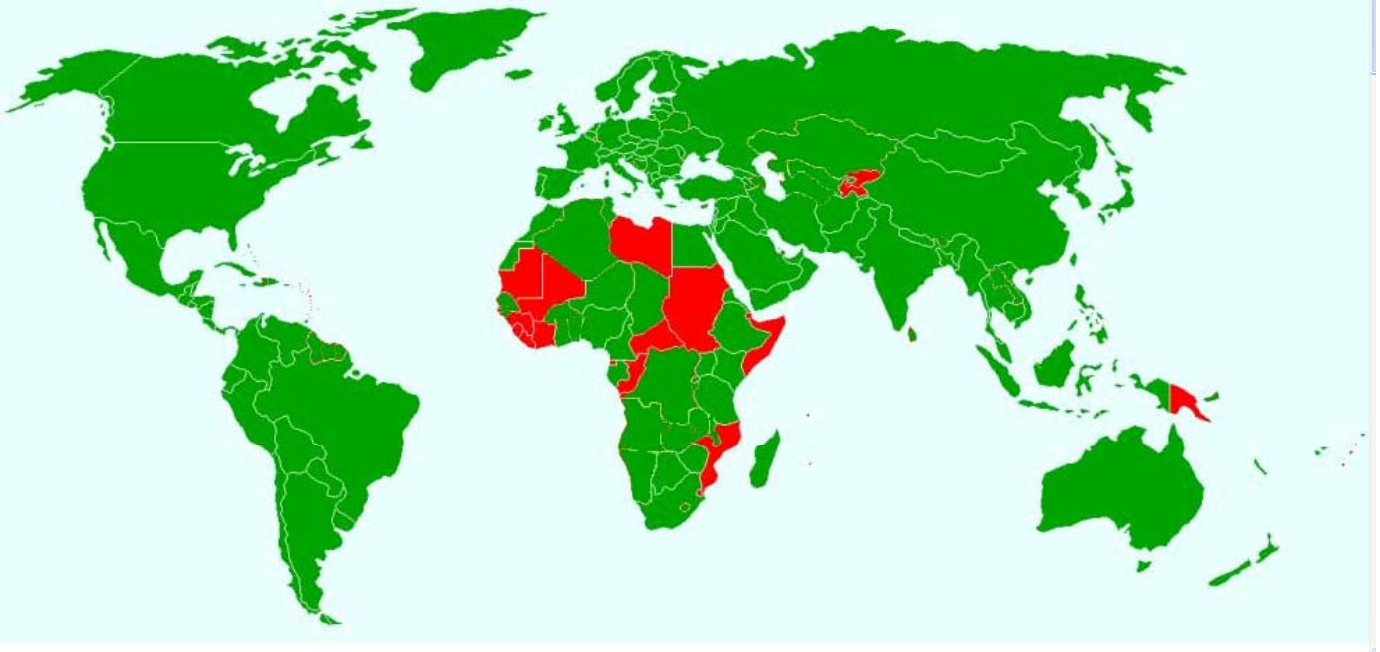
<table>
<tr><td>Textual</td><td>$(\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$
$\delta = \{(q_0, b, q_0), (q_0, a, q_1), (q_1, a, q_0), (q_1, b, q_2), (q_2, a, q_1)\}$</td></tr>
<tr><td>Tabular</td><td><table>
<tr><td></td><td>a</td><td>b</td></tr>
<tr><td>q_0</td><td>q_1</td><td>q_0</td></tr>
<tr><td>q_1</td><td></td><td>q_2</td></tr>
<tr><td>q_2</td><td></td><td></td></tr>
</table></td></tr>
<tr><td>Visual</td><td> 
</td></tr>
<tr><td>Interactive</td><td> 
</td></tr>
</table>

Overview of JFLAP

- **J**ava **F**ormal **L**anguages and **A**utomata **P**ackage
- 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!

JFLAP's Use Around the World

- JFLAP web page has over 200,000 hits since 1996
- Google Search
  - JFLAP appears on over 9830 web pages
  - Note: search only public web pages
- JFLAP now used in several textbooks – JFLAP exercises
- JFLAP been downloaded in over 160 countries



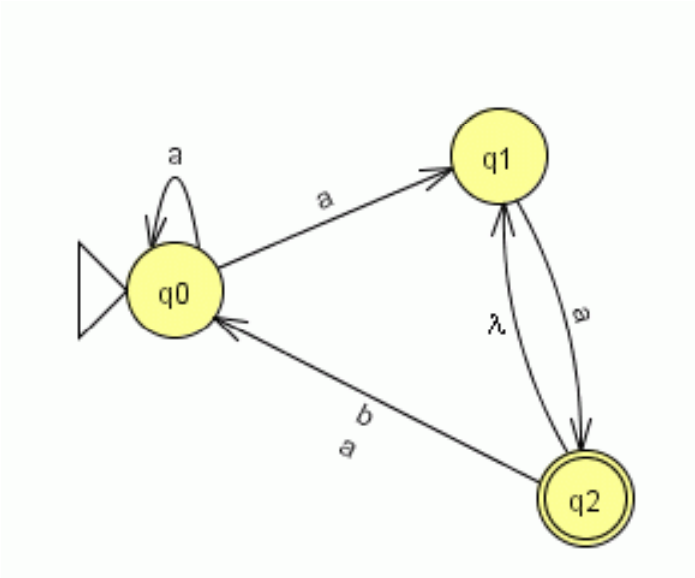
Thanks to Students - Worked on JFLAP and Automata Theory Tools

Over 19 years!

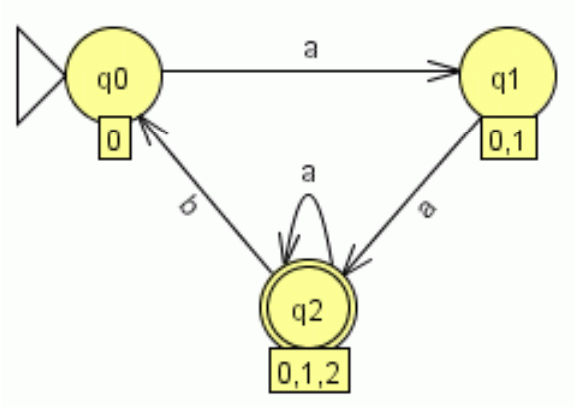
- NPDA - 1990, C++, Dan Caugherty
- FLAP - 1991, C++, Mark LoSacco, Greg Badros
- JFLAP - 1996-1999, Java version
  Eric Gramond, Ted Hung, Magda and Octavian Procopiuc
- Pâté, JeLLRap, Lsys
  Anna Bilska, Jason Salemme, Lenore Ramm, Alex Karweit, Robyn Geer
- JFLAP 4.0 – 2003, Thomas Finley, Ryan Cavalcante
- JFLAP 6.0 – 2005-2008 Stephen Reading, Bart Bressler, Jinghui Lim, Chris Morgan, Jason Lee
- JFLAP 7.0 - 2009 Henry Qin, Jonathan Su

What is JFLAP?
Regular Languages

- Create
  - DFA and NFA
  - Moore and Mealy
  - regular grammar
  - regular expression

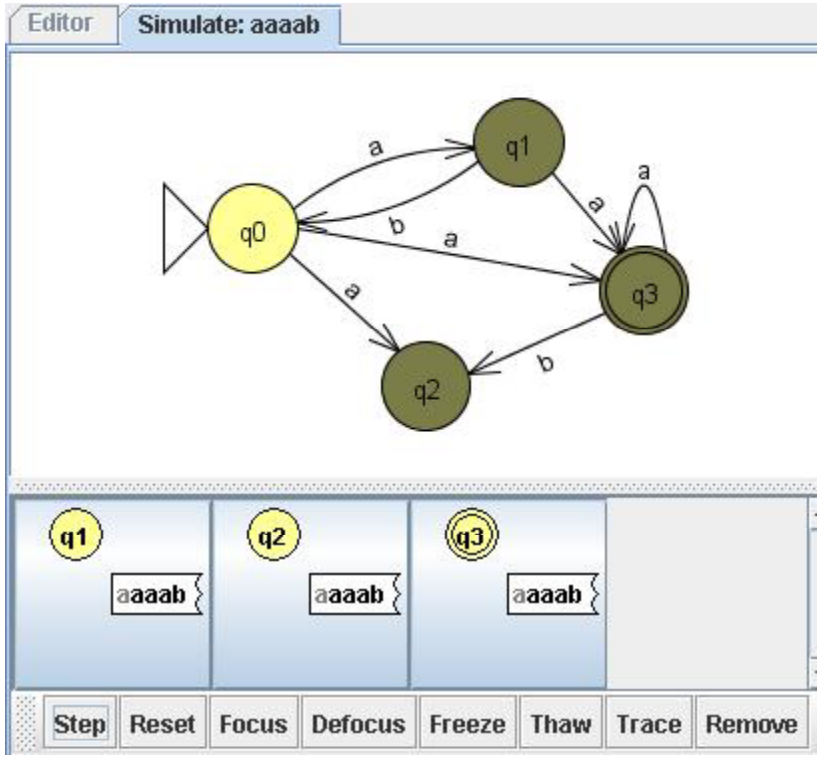


- Conversions
  - NFA to DFA to minimal DFA
  - NFA ←→ regular expression
  - NFA ←→ regular grammar



JFLAP – Regular languages (more)

- Simulate DFA and NFA
  - Step with Closure or Step by State
  - Fast Run
  - Multiple Run
- Combine two DFA
- Compare Equivalence
- Brute Force Parser
- Pumping Lemma

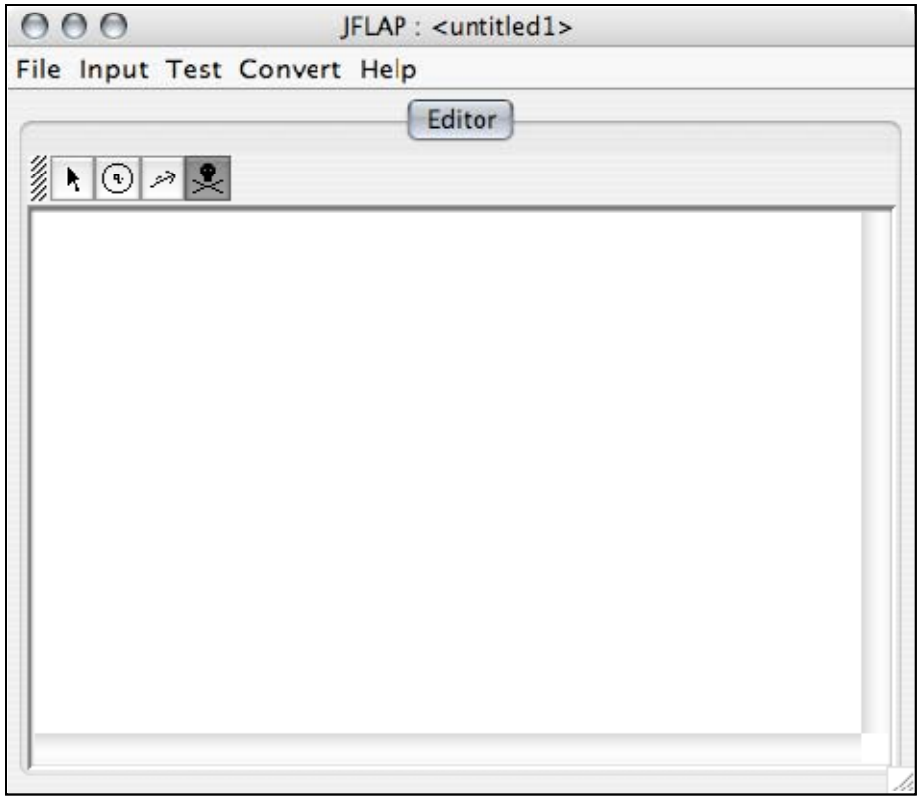


FA Edit & Simulation
Start up JFLAP

- When we start up JFLAP we have a choice of structures.
- The first of these is the Finite Automata!

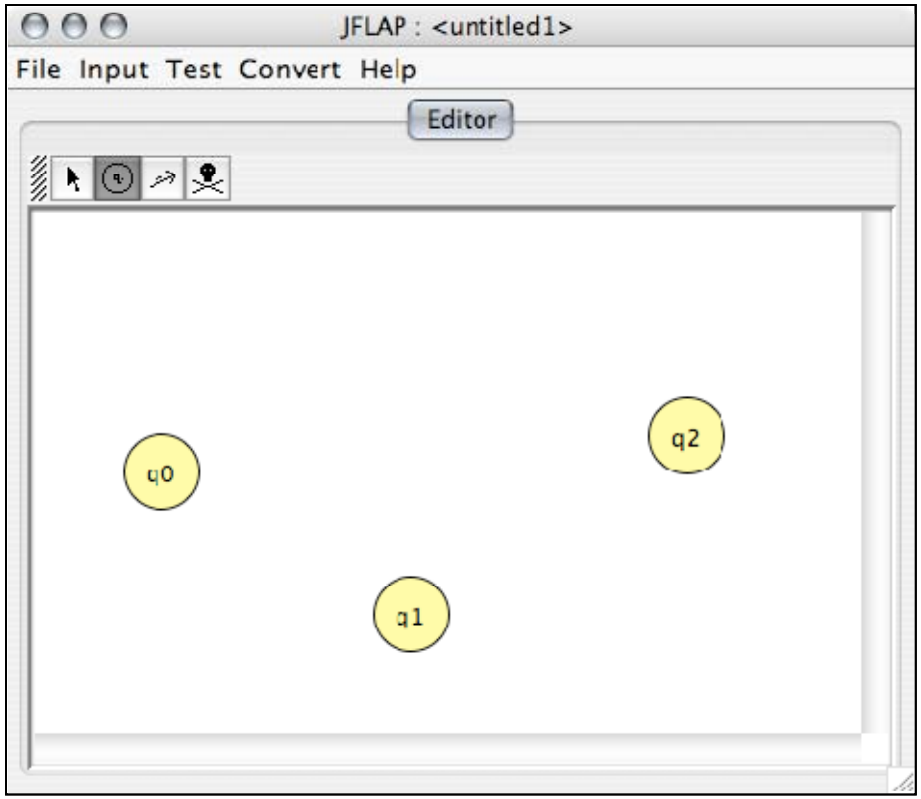
FA Edit & Simulation
Start Editing!

- We start with an empty automaton editor window.



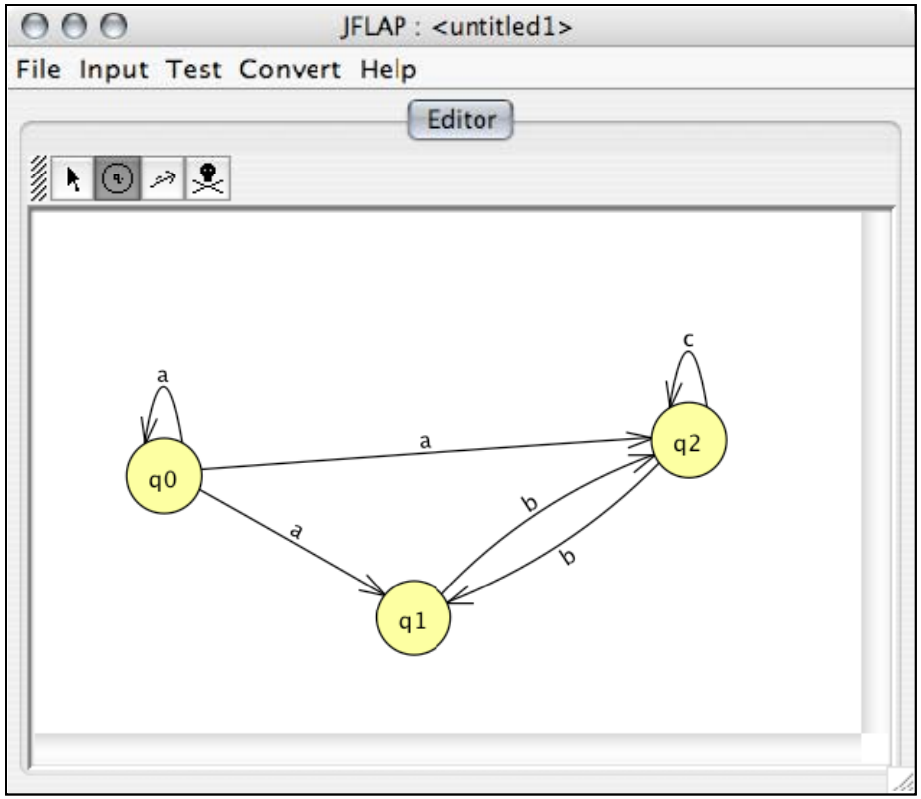
FA Edit & Simulation
Create States

- We create some states ...



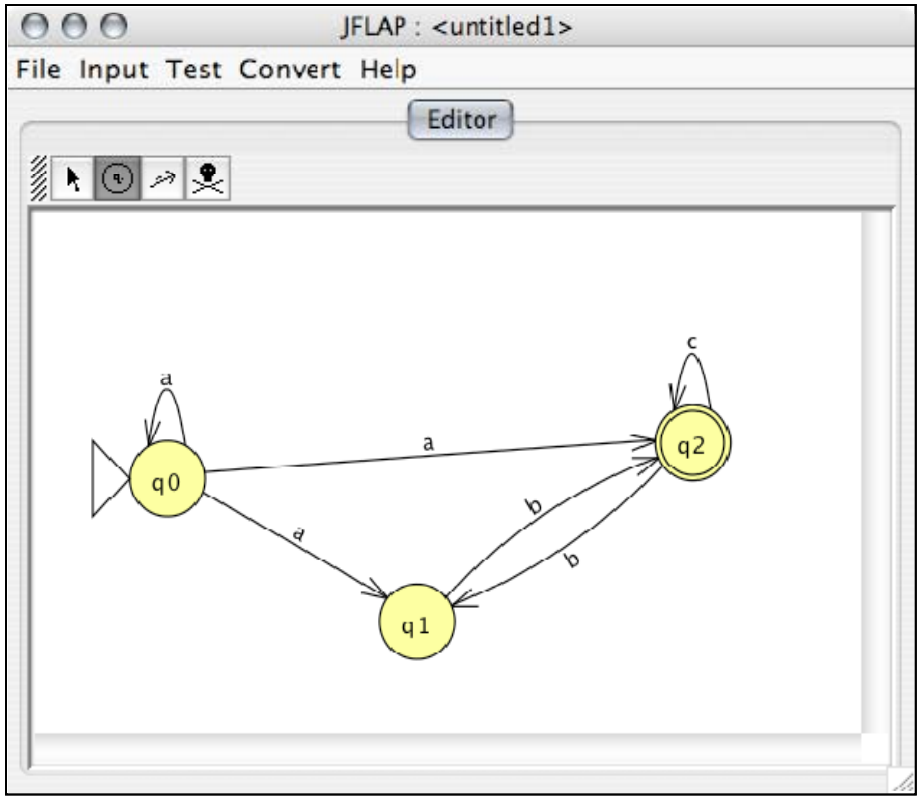
FA Edit & Simulation
Create Transitions

- We create some transitions ...



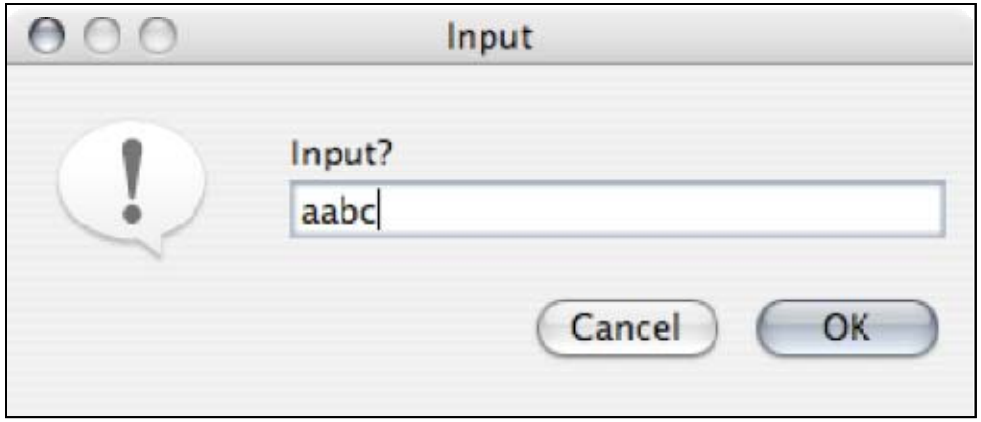
FA Edit & Simulation
Initial and Final State

- We set an initial and final state.
- Now we can simulate input on this automaton!



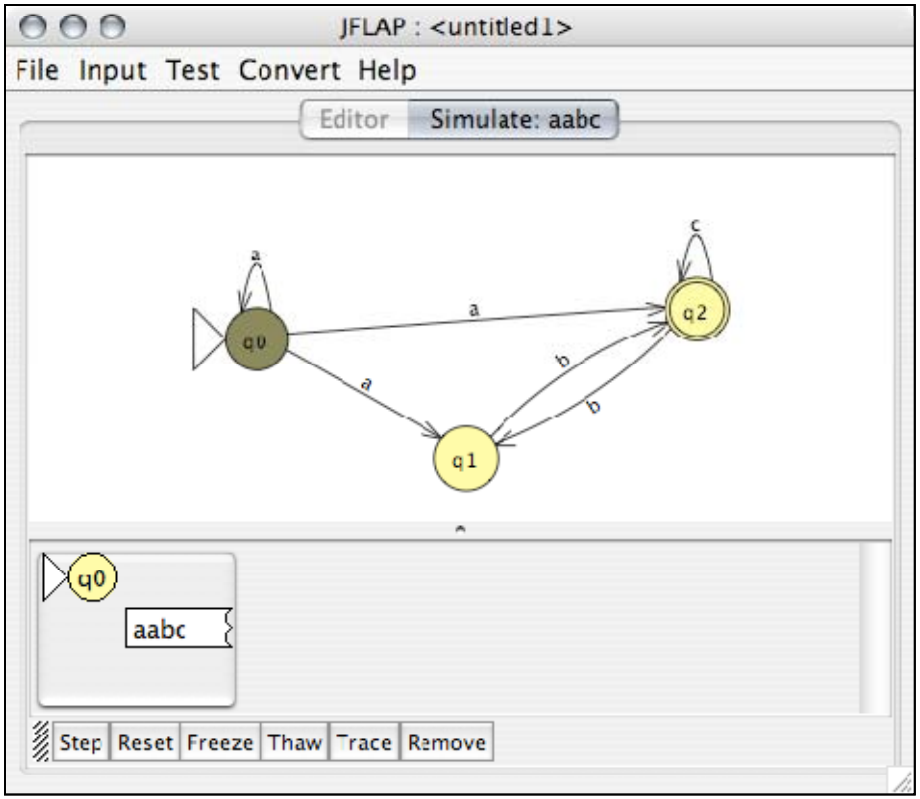
FA Edit & Simulation
Input to Simulate...

- When we say we want to simulate input on this automaton, a dialog asks us for the input.



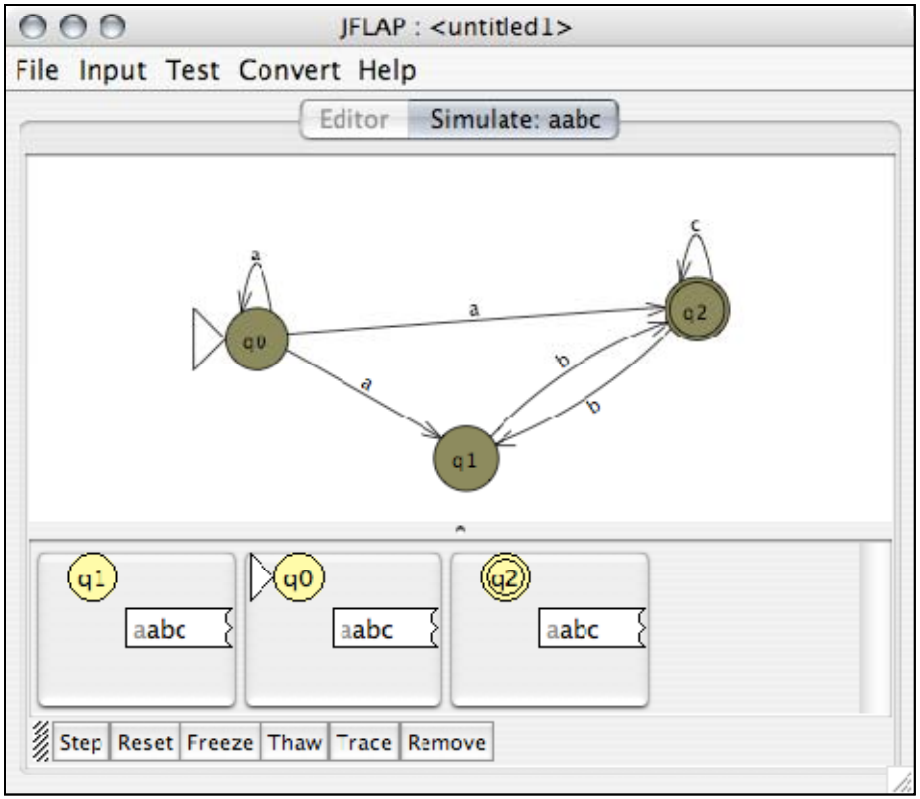
FA Edit & Simulation
Start Simulation!

- When simulation starts, we have a configuration on the initial state with all input remaining to be processed.



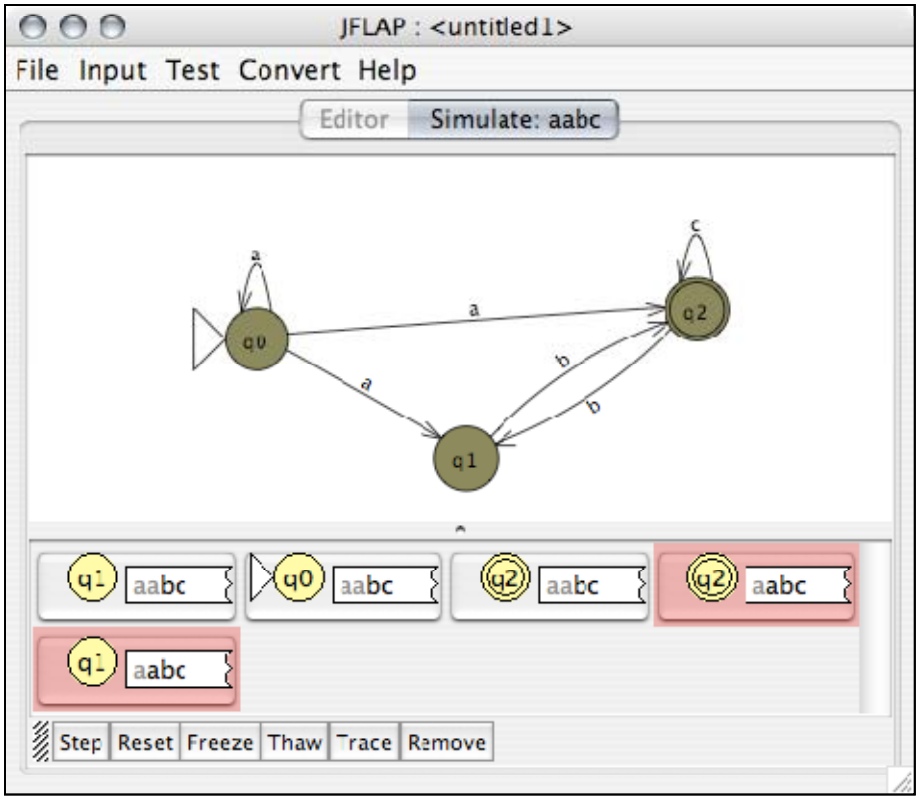
FA Edit & Simulation After One Step

- This is a nondeterministic FA, and on this input we have multiple configurations after we “Step.”



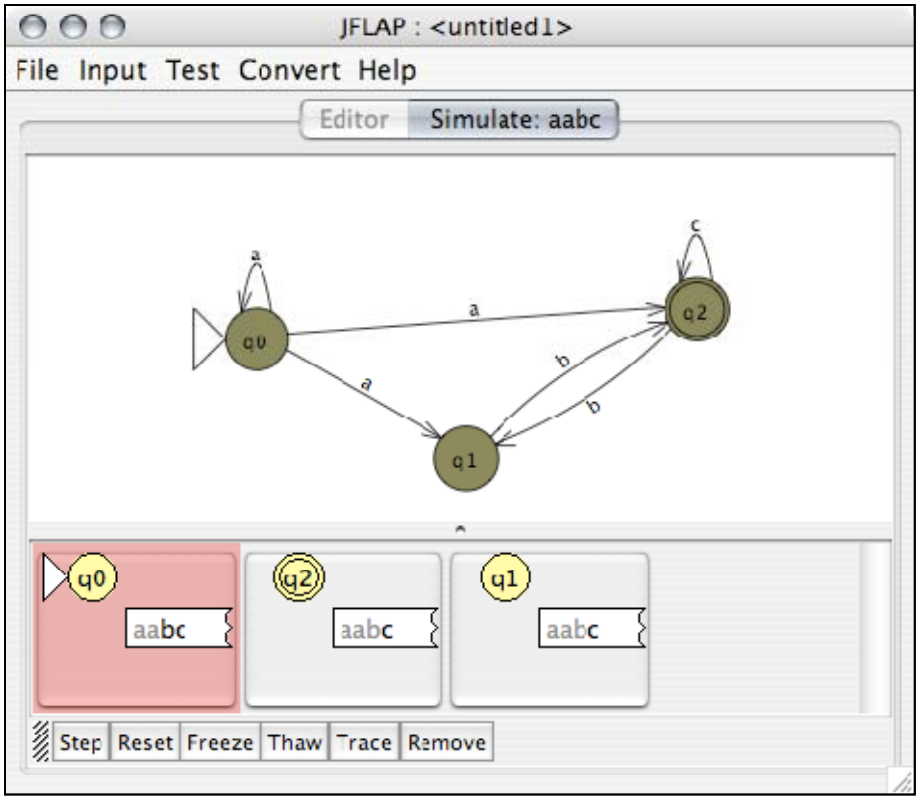
FA Edit & Simulation After Two Steps

- The previous configurations on q_1 and q_2 are rejected, and are shown in red.
- The remaining uncolored configurations paths are not rejected, and are still open.



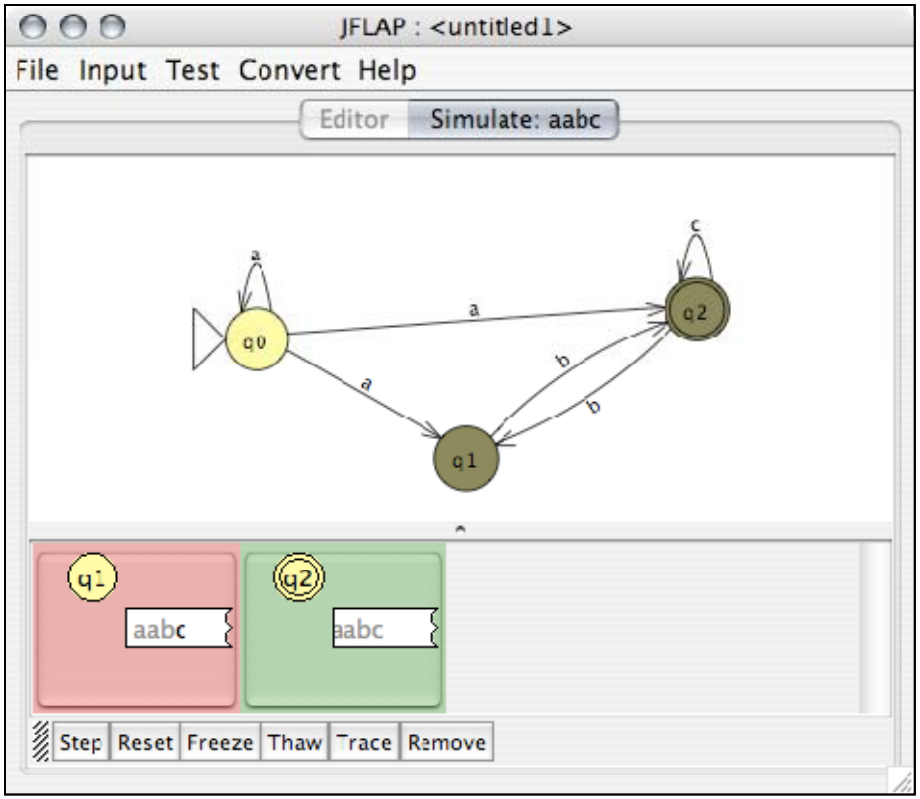
FA Edit & Simulation After Three Steps

- Yet another step.



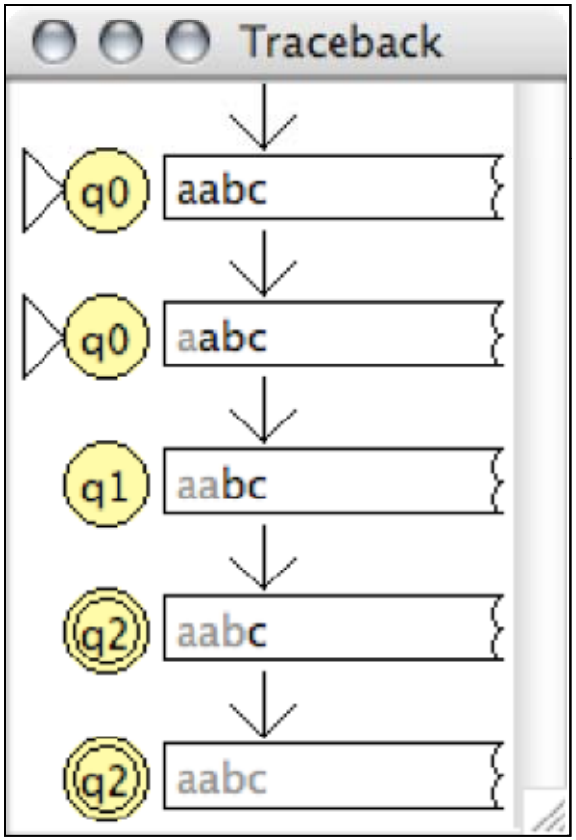
FA Edit & Simulation After Four Steps

- One of the final configurations has been accepted!



FA Edit & Simulation Traceback

- One can then see a traceback to see the succession of configurations that led to the accepting configuration.



FA Multiple Run

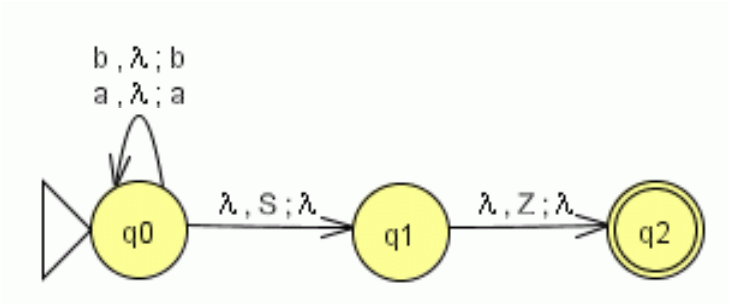
- Select Multiple Run
- One can then enter many strings and receive acceptance info.

Multiple Run

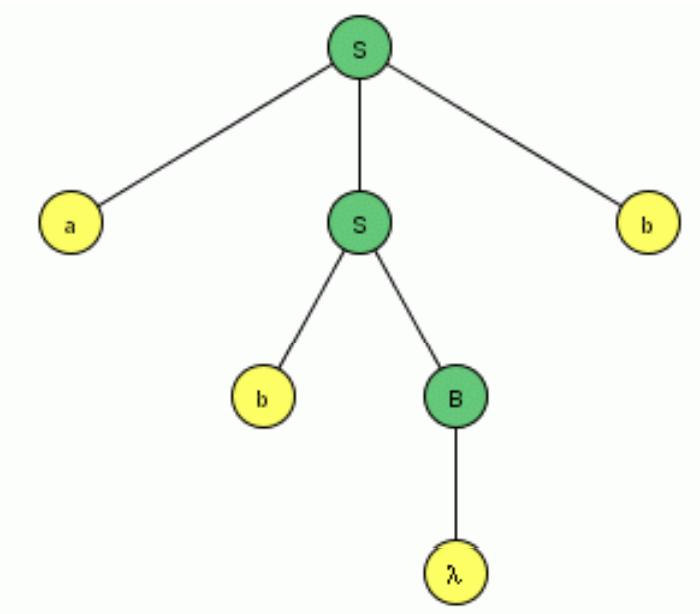
Input	Result
a	Accept
aa	Accept
aab	Accept
aabb	Accept
	Reject
acb	Reject
abcbb	Accept
abbcc	Accept
abcab	Reject
bc	Reject

JFLAP – Context-free Languages

- Create
  - Nondeterministic PDA
  - Context-free grammar
  - Pumping Lemma

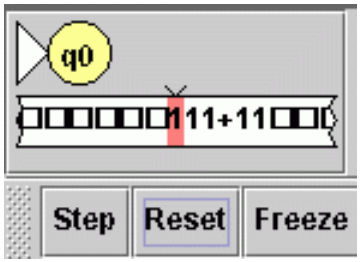


- Transform
  - PDA → CFG
  - CFG → PDA (LL & SLR parser)
  - CFG → CNF
  - CFG → Parse table (LL and SLR)
  - CFG → Brute Force Parser

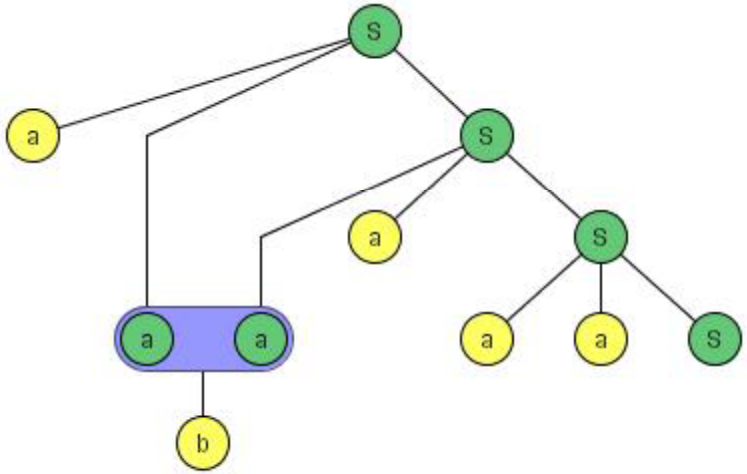


JFLAP – Recursively Enumerable Languages

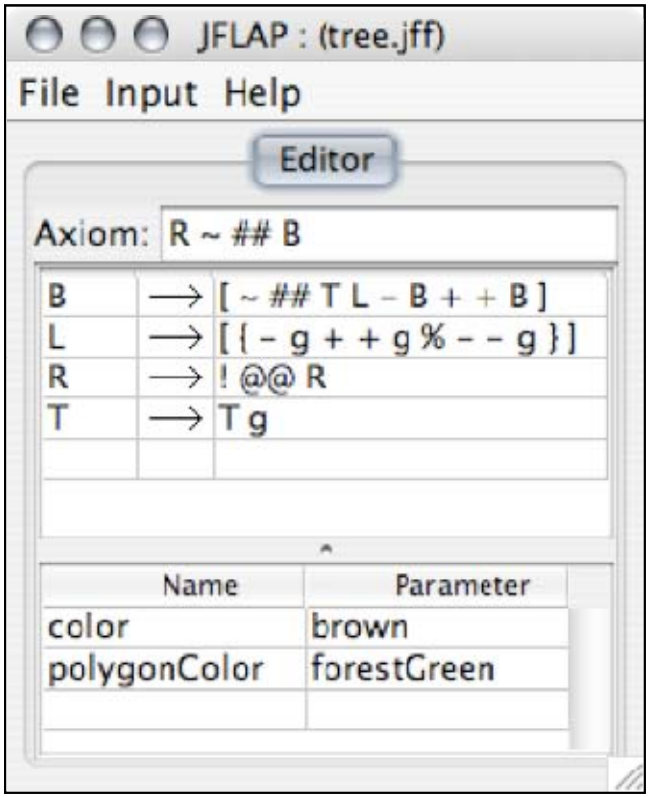
- Create
  - Turing Machine (1-Tape)
  - Turing Machine (multi-tape)
  - Building Blocks
  - Unrestricted grammar



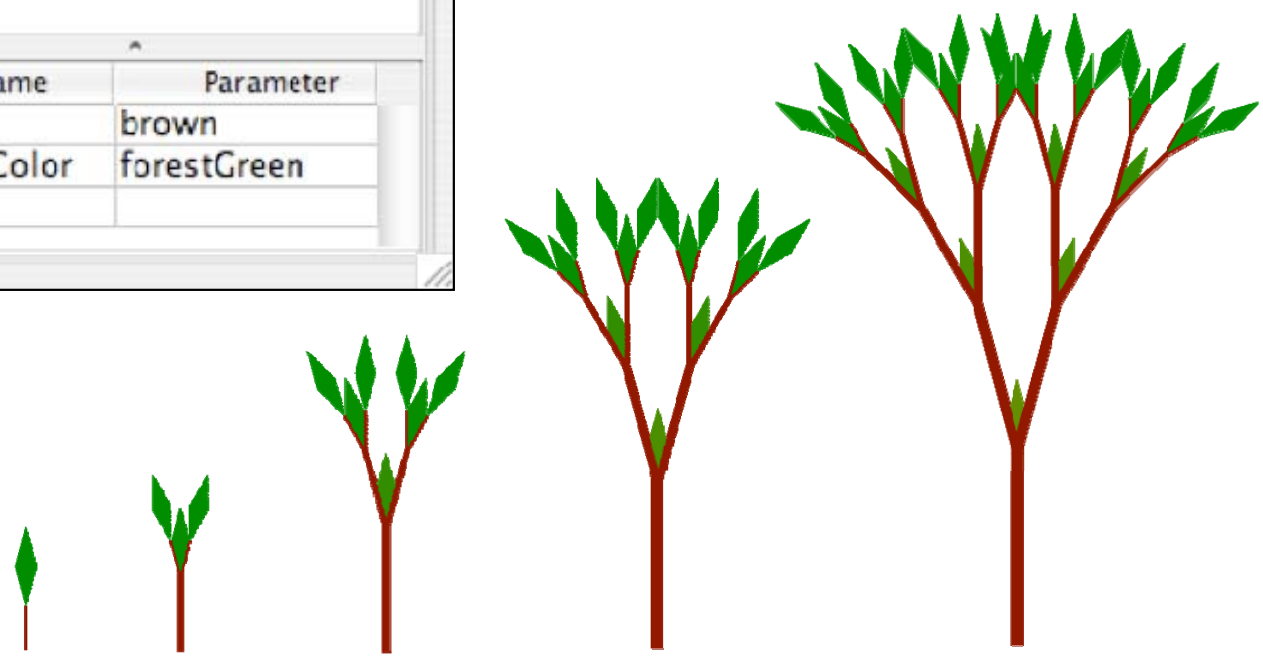
- Parsing
  - Unrestricted grammar with brute force parser



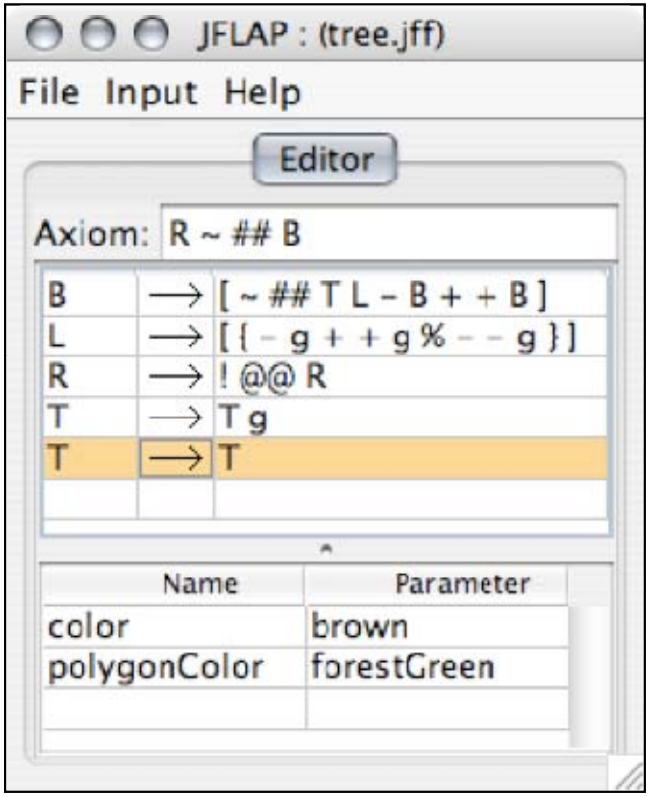
JFLAP - L-Systems



- This L-System renders as a tree that grows larger with each successive derivation step.



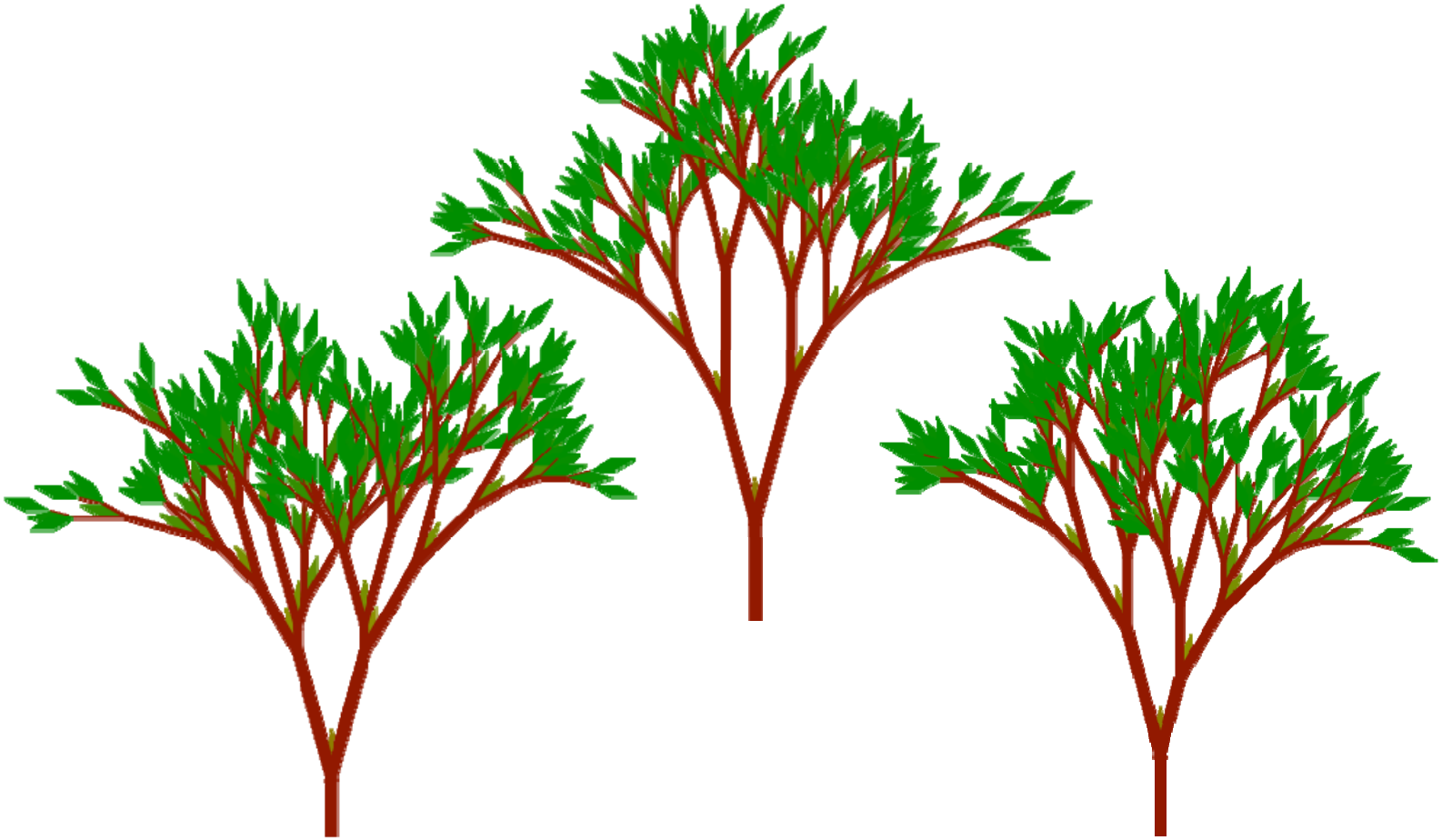
L-Systems



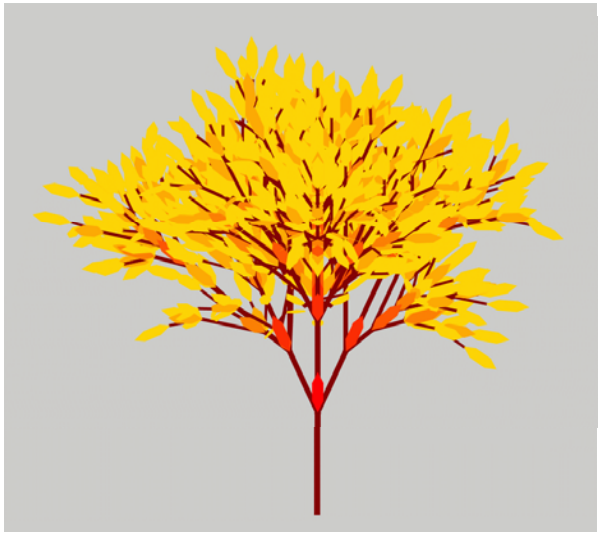
- L-systems may also be stochastic.
- The $T \rightarrow Tg$ rule adds g to the derivation, which draws a line segment.
- We add another rewriting rule for T, $T \rightarrow T$.
- With two rewriting rules for T, the rule chosen is random, leading to uneven growth!

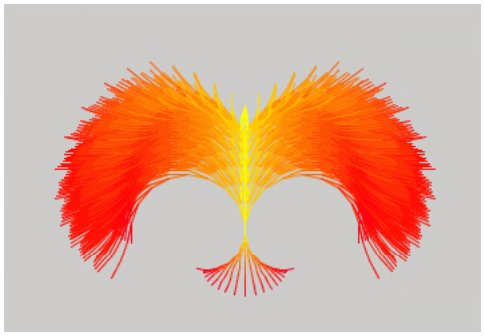
L-Systems

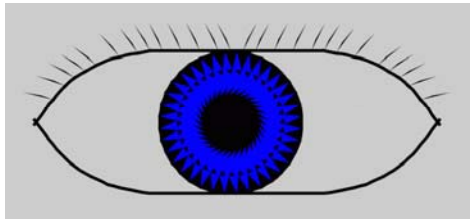
The same stochastic L-system, rendered 3 different times all at the 9th derivation.

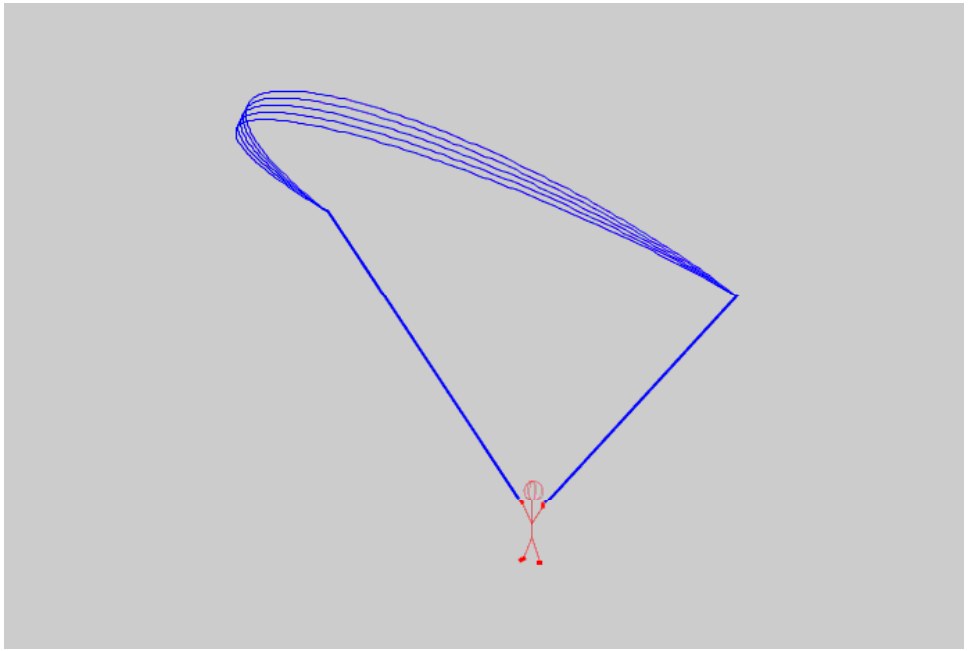


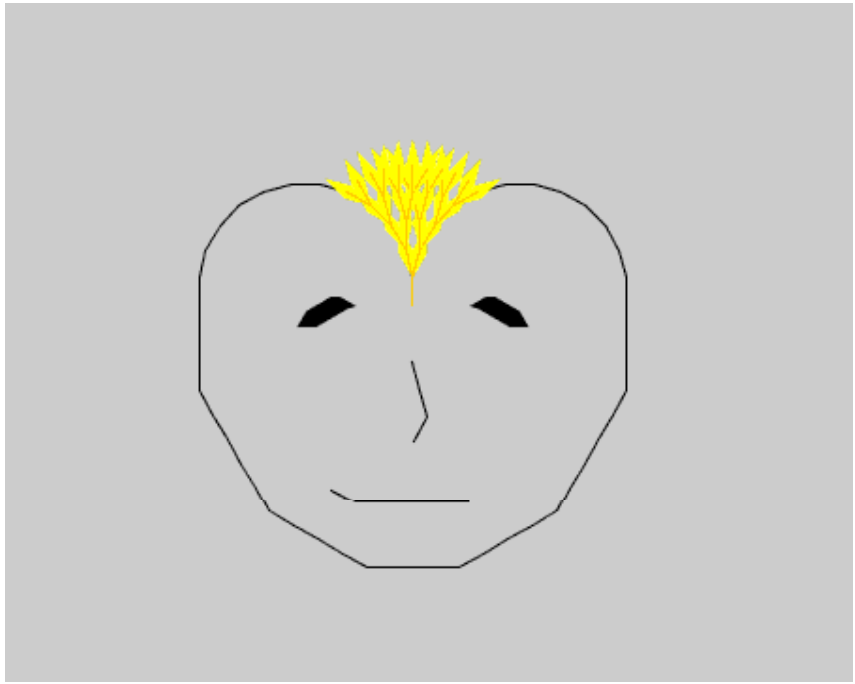
Students love L-Systems











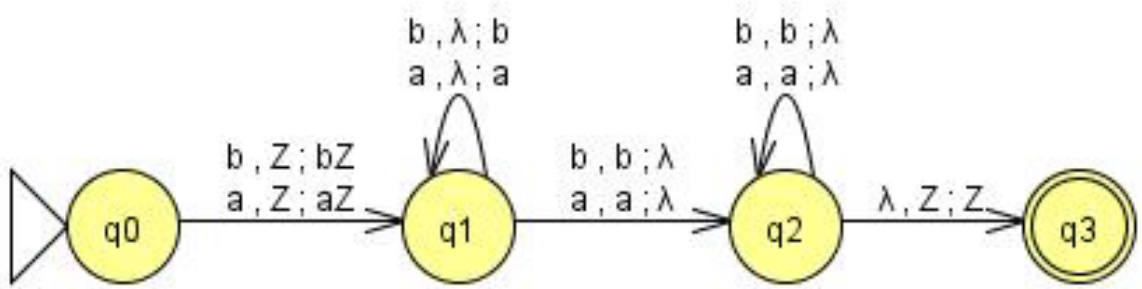
Increasing Interaction in the Course with JFLAP

Using JFLAP during Lecture

- Use JFLAP to build examples of automata or grammars
- Use JFLAP to demo proofs
- Load a JFLAP example and students work in pairs to determine what it does, or fix it if it is not correct.

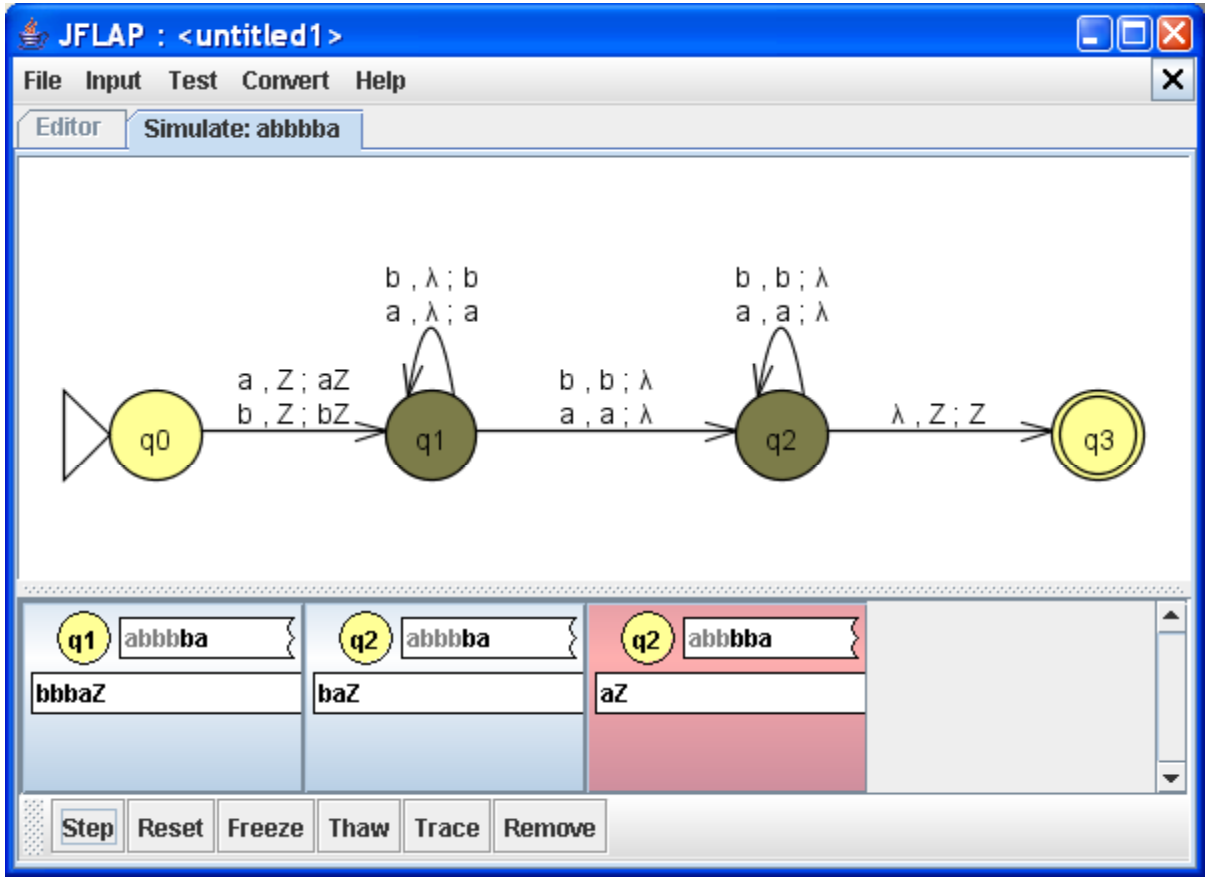
Example 1: JFLAP during Lecture

- Ask students to write on paper an NPDA for palindromes of even length
- Build one of their solutions using JFLAP
  - Shows students how to use JFLAP



Example 1: JFLAP during Lecture (cont)

- Run input strings on the NPDA
  - Shows the nondeterminism



Example 2: JFLAP during Lecture

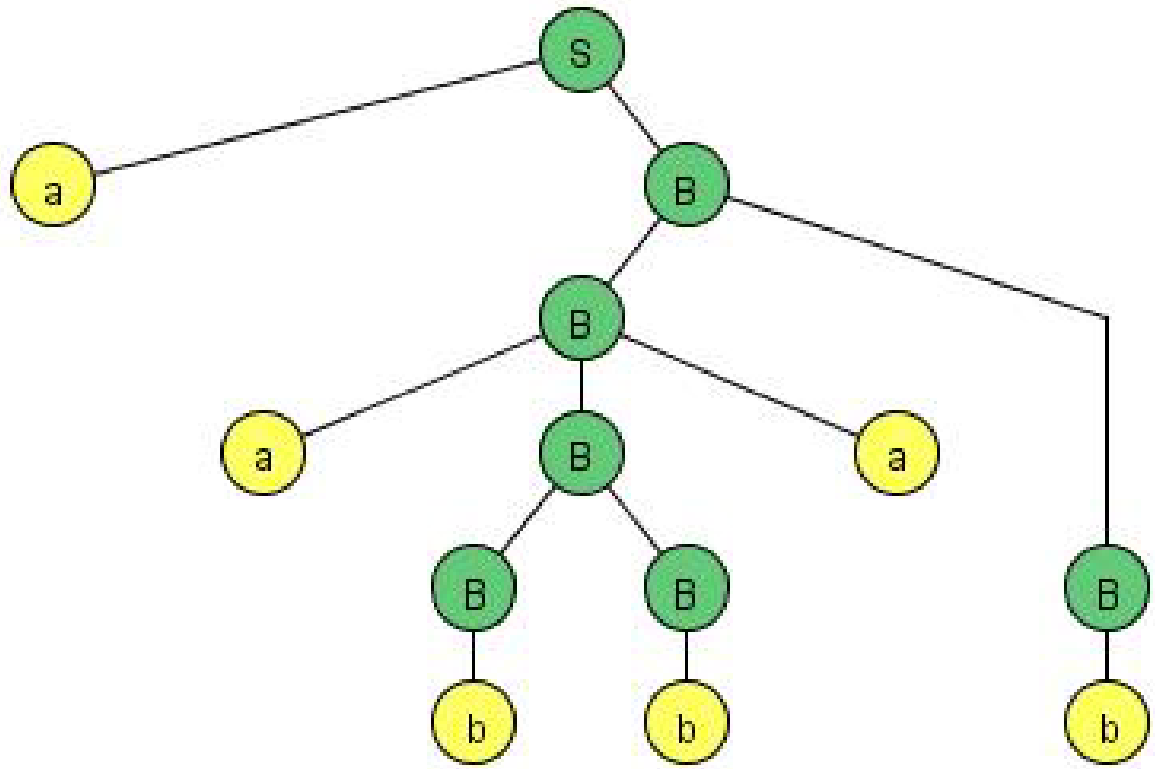
- Brute Force Parser
  - Give a grammar with a lambda-production and unit production
  - Run it in JFLAP, see how long it takes (LONG)
    - Is aabbab in L?
  - Transform the grammar to remove the lambda and unit-productions
  - Run new grammar in JFLAP, runs much faster!

S	→	aB
B	→	BB
B	→	aBa
B	→	b
B	→	λ

S	→	aB
B	→	BB
B	→	aBa
B	→	b
S	→	a
B	→	B
B	→	aa

Example 2 (cont)
Parse Tree Results

- First Grammar – 1863 nodes generated
- Second Grammar – 40 nodes generated
- Parse tree is the same.



With JFLAP, Exploring Concepts too tedious for paper

- Load a Universal Turing Machine and run it
- See the exponential growth in an NFA or NPDA
- Convert an NPDA to a CFG
  - Large grammar with useless rules
  - Run both on the same input and compare
  - Transform grammar (remove useless rules)

JFLAP's use Outside of Class

- Homework problems
  - Turn in JFLAP files
  - OR turn in on paper, check answers in JFLAP
- Recreate examples from class
- Work additional problems
  - Receive immediate feedback

Ordering of Problems in Homework

- Order questions so they are incremental in the usage of JFLAP

1. Load a DFA. What is the language?

   Students only enter input strings.

2. Load a DFA that is not correct. What is wrong? Fix it.

   Students only modifying a small part.

3. Build a DFA for a specific language.

   Last, students build from scratch.

JFLAP Study

- Study of JFLAP's effectiveness in learning
  - Two year study
  - Fourteen Faculty Adopters
  - Two 2-day faculty Adopter Workshops – June 2005, June 2006
  - Collect data 2005-06 and 2006-07 Academic years
  - Pretest/Posttest
  - Interviews
  - Team of three evaluators
    - Eric Weibe – Education
    - Rocky Ross – Computer Science Theory
    - Joe Bergin – Computer Science Tools

Fourteen Faculty Adopter Participants

-small, large
- public, private
- includes minority institutions

- Duke
- UNC-Chapel Hill
- Emory
- Winston-Salem State University
- United States Naval Academy
- Rensselaer Polytechnic Institute
- UC Davis
- Virginia State University
- Norfolk State University
- University of Houston
- Fayetteville State University
- University of Richmond
- San Jose State University
- Rochester Institute of Technology

We hoped to show with this learning approach…

- Students gain a better and deeper understanding of FLA
- Students are happier and more confident in learning FLA
- Students are more interested in using the tools on their own
- Instructors can easily use the tools in class
- Instructors can easily grade electronic submissions

Running a Study is hard!

- Hit by the drop In enrollments in CS after dot-com burst
- IRBs are different process at every institution
  - One page writeup ok'd (simplest)
  - Full medical IRB (many pages)
- One institution shut down all IRB research projects – we could not use data already collected.
- One University - Control Group – different times means different types of students, different professors.
- Some faculty came to workshop and did not follow through
- There were also some fantastic faculty!
- A lot depends on the people you pick to participate!

Second Year

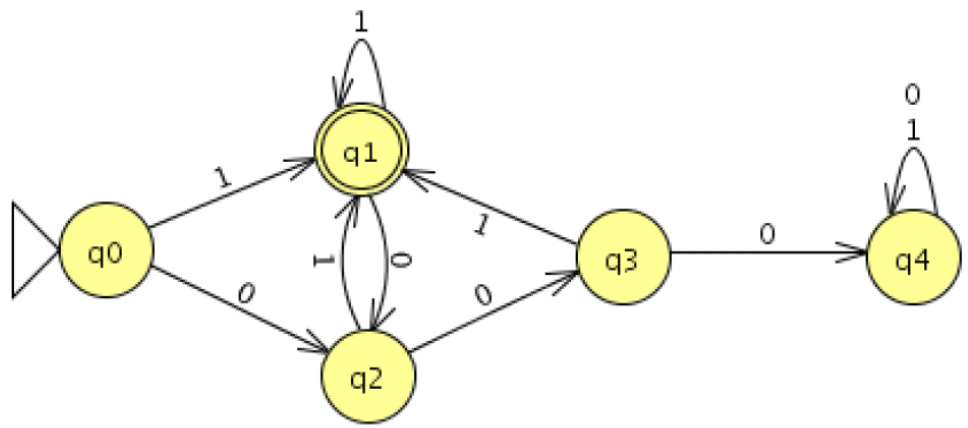
- Added two schools with large number of students
  - One school worked well
    - Multiple sections – one without JFLAP
    - One class in morning, one in evening – different type of students
  - One school did not follow through

Do Student's Learn Better with JFLAP?

- Pretest/Posttest
  - No statistically significant difference between experimental and control groups.

PROBLEM 1:

What is the language of the following finite automaton over the alphabet $\Sigma = \{0, 1\}$?



A. All strings that end in 1

B. All strings that end in 1, 01 or 001

C. All strings such that there cannot be more than 2 adjacent 0's

D. All strings that end in 1 and there cannot be more than 2 adjacent 0's

E. Not familiar with the topic

Year One Instructor Interviews

- Used JFLAP in their courses
  - Primary use in class – demonstrations
  - Some used it to generate the graphics for their lecture
  - Extensive use – homeworks – includes electronic submission
  - One used it in office hours

Year One – Software Implementation

Question	YES/NO	No.	%
Did you use JFLAP software to study for inclass exams?	YES NO	20 16	55% 45%
Did you feel you had time to learn how to use the JFLAP software?	YES NO	33 2	94% 6%
Did you feel that using the software took time away from other study activities?	YES NO	3 33	8% 92%
Did the time and effort it took to use JFLAP help you get a better grade in the course?	YES NO	23 13	64% 36%
Was it easier to use JFLAP software or was it easier to draw it out by hand?	software by hand	30 6	83% 17%
Did you feel you would have done as well in the course if you had not used JFLAP?	YES NO NA	18 13 5	50% 36% 14%

Years 1 and 2: Usability Survey

Question		
How easy was it to use the drawing tool of JFLAP? (134 respondents)	Very Easy	31%
	Easy	48%
	Neither	15%
	Difficult	5%
	Very Difficult	0%
How easy was it to run the automata you designed in JFLAP? (134 respondents)	Very easy	33%
	Easy	47%
	Neither	12%
	Difficult	6%
	Very Difficult	2%
How easy was it to interpret results from the test run in JFLAP? (134 respondents)	Very Easy	23%
	Easy	45%
	Neither	19%
	Difficult	10%
	Very Difficulty	3%
What is your overall assessment of the JFLAP software? (133 respondents)	Very Poor	2%
	Poor	4%
	Neither	11%
	Good	63%
	Very Good	20%

Year 2 – Implementation Survey

QUESTION	Time	Response
When preparing for exams what percentage of study time involved the use of JFLAP software? (100 responses)	0-20%	68%
	21-40%	16%
	41-60%	11%
	61-80%	3%
	81-100%	2%
How often did you use JFLAP to do additional practice problems? (99 responses)	Never	46%
	Rarely	25%
	Occasionally	21%
	Often	4%
	Very Often	4%

Year 2 – Usability Survey

Question		
Using JFLAP made the course more enjoyable for me. (98 responses)	Strongly Agree	12%
	Agree	51%
	Neither	25%
	Disagree	6%
	Strongly Disagree	4%
Using JFLAP made me feel more engaged in the course. (98 responses)	Strongly Agree	13%
	Agree	59%
	Neither	15%
	Disagree	9%
	Strongly Disagree	3%
Having access to JFLAP made learning course concepts ... (97 responses)	Much harder	1%
	Harder	5%
	Neither	26%
	Somewhat easier	54%
	Much easier	14%

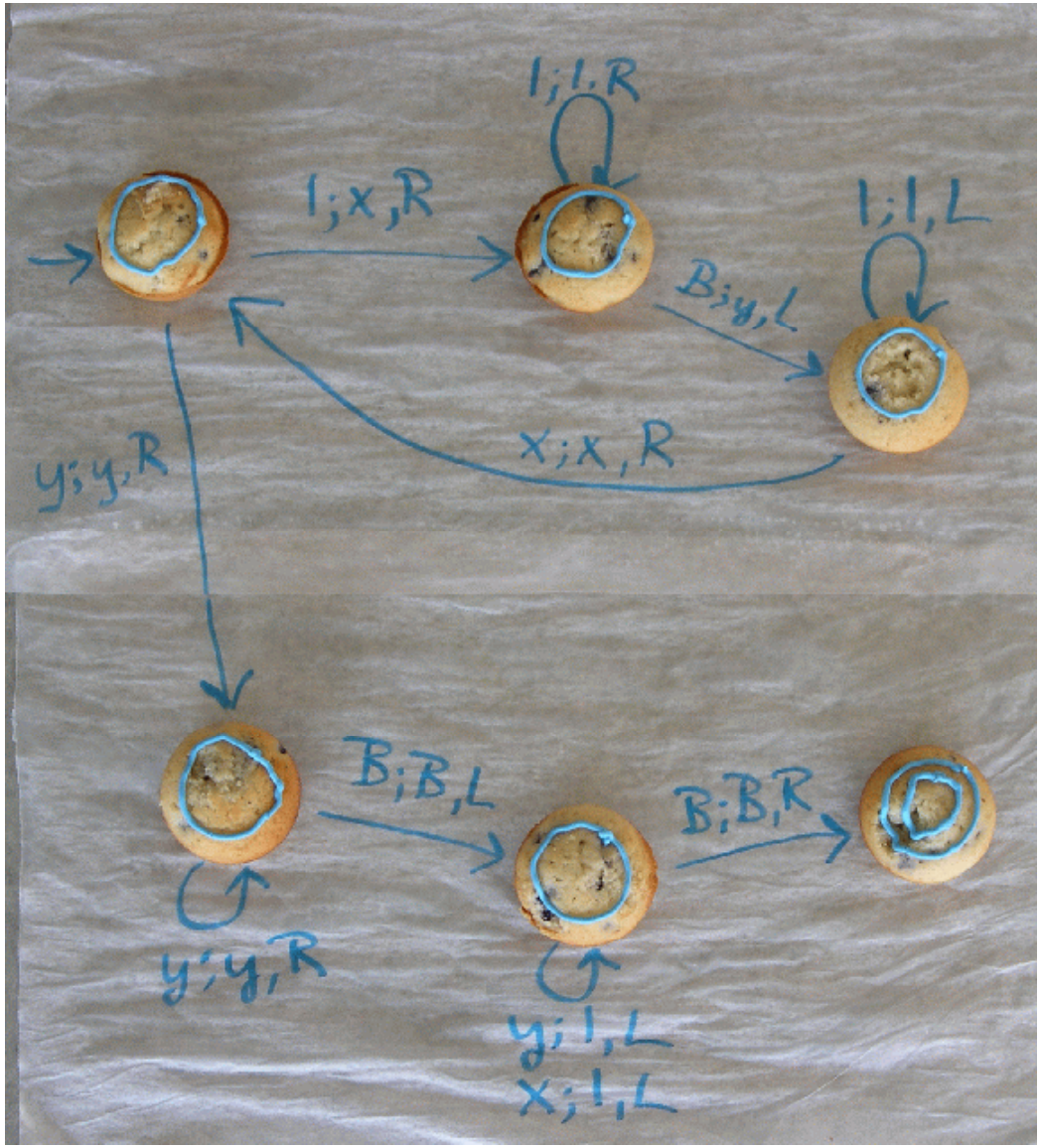
Conclusions From Study

- No Conclusive Results from Pretests/Postests
- Results of Study showed
  - All the faculty used JFLAP in their courses, mostly for homework, some in lecture
  - Students had a high opinion of JFLAP
  - Majority of students felt access to JFLAP
    - Made learning course concepts easier
    - Made them feel more engaged
    - Made the course more enjoyable
  - Over half the students used JFLAP to study for exams
  - Over half the student thought time and effort using JFLAP helped them get a better grade.

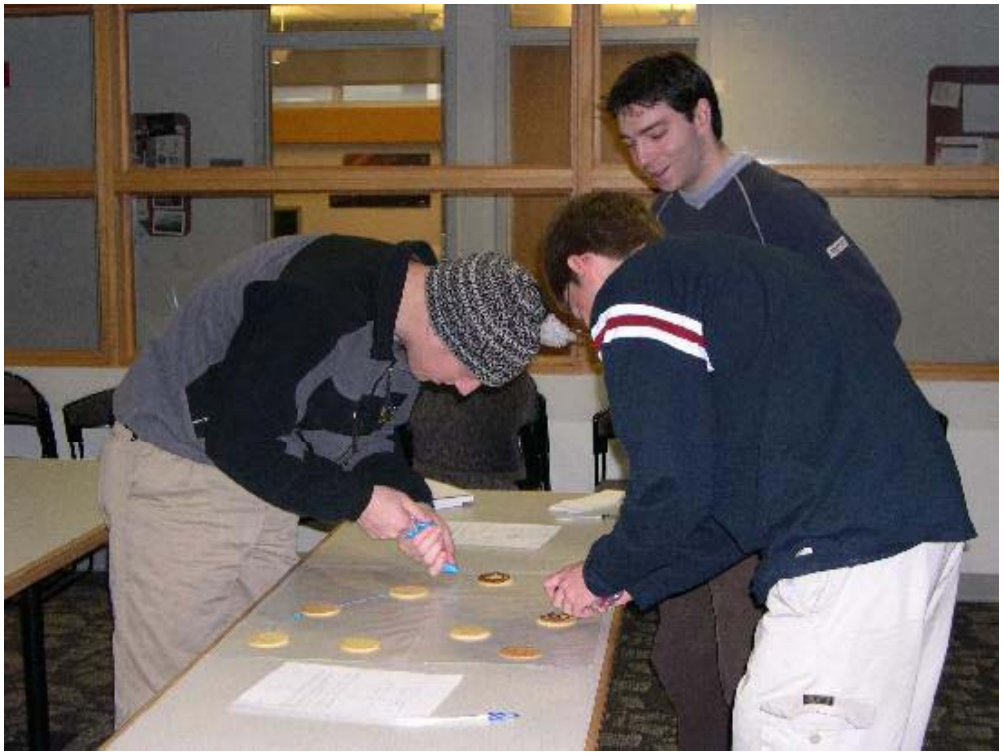
There are other ways to get interaction in this course besides software…

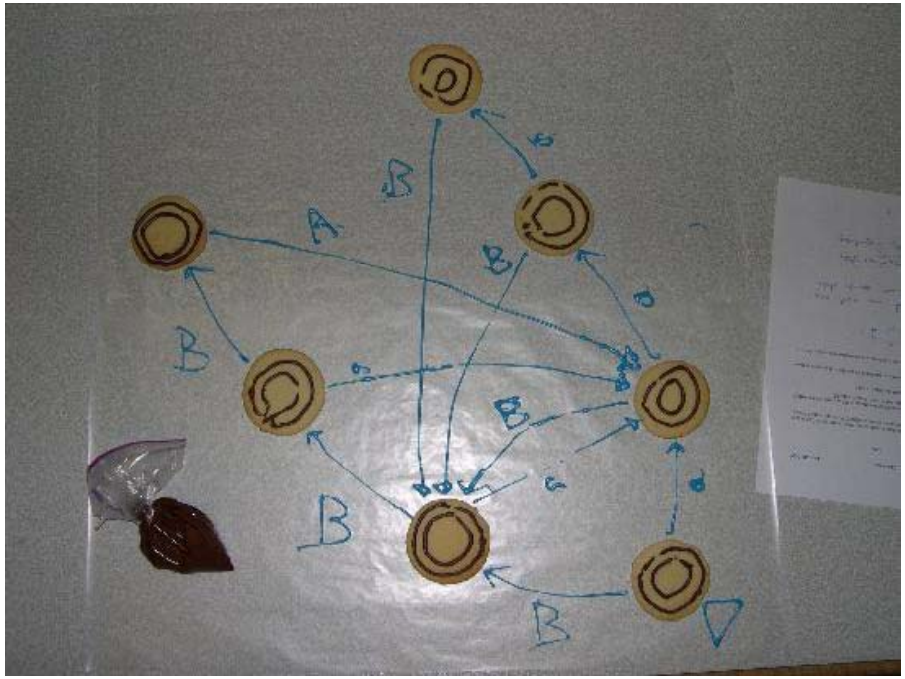
Interaction in Class – Props
Edible Turing Machine

- TM for f(x)=2x where x is unary
- TM is not correct, can you fix it? Then eat it!
- States are blueberry muffins

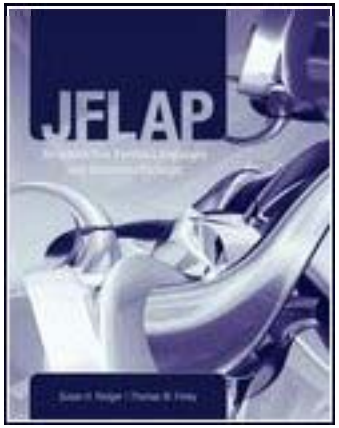


Students building DFA with cookies and icing



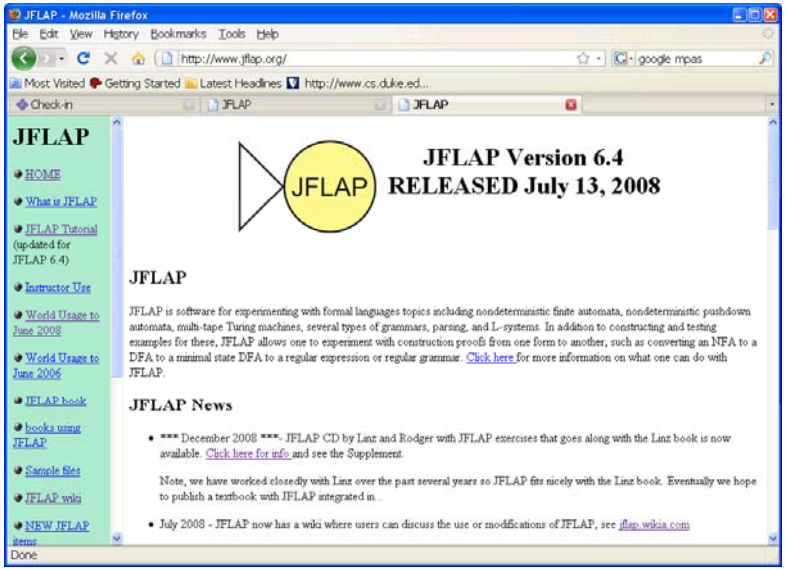


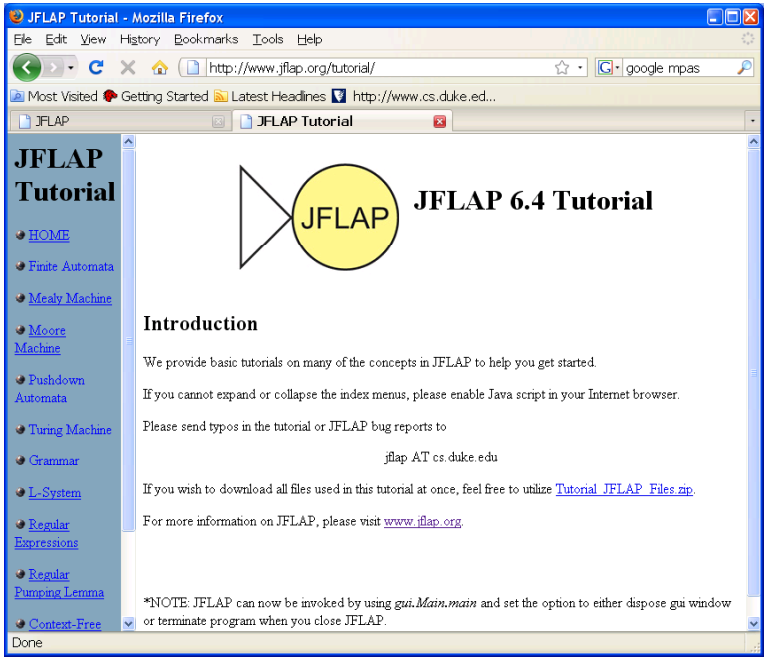
JFLAP



www.jflap.org

JFLAP tutorial





JAWAA
Java and Web-based Algorithm Animation

- Scripting Language for Animation
- Easily create, modify and move objects
- Runs over the web, no need to install
- More Advanced Students
  - Output JAWAA Command from Program
  - Animate Data Structures Easily
- SIGCSE 2003 and SIGCSE 1998
- www.cs.duke.edu/~rodger/tools/
- Students: Pierson, Patel, Finley, Akingbade, Jackson, Gibson, Gartland

Related Work

- Samba, Jsamba - Stasko (Georgia Tech)
- AnimalScript – Roessling (Darmstadt Univ of Tech, SIGCSE 2001)
- JHAVE – Naps (U. Wisc. Oshkosh, SIGCSE 2000)
- Lots of animations and systems on the web!

JAWAA Commands

`circle c1 30 20 60 blue red`	
`moveRelative c1 60 0`	move right
`moveRelative c1 0 50`	move down
`changeParam c1 bkgrd blue`	

JAWAA Primitives

circle	
rectangle	
line	
oval	
polygon	
text	jawaa

JAWAA Data Structures

Array

```
array people 25 25 4.2 Owen running Gail boating
   Robert toys Susan cakes vert red yellow black
changeParam people index on
changeParam people[1] bkgrd white
changeParam people[0].1 text bubblesort
moveRelative people[2] 30 0
changeParam people[2] swap people[0]
```

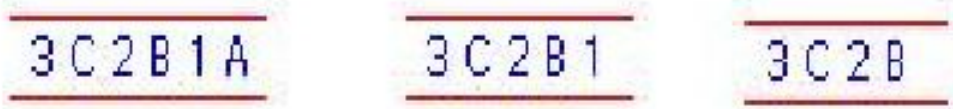
JAWAA Data Structures

- Stack

```
stack s1 200 200 4 Pop The Top Off black red
pop s1
pop s1
```

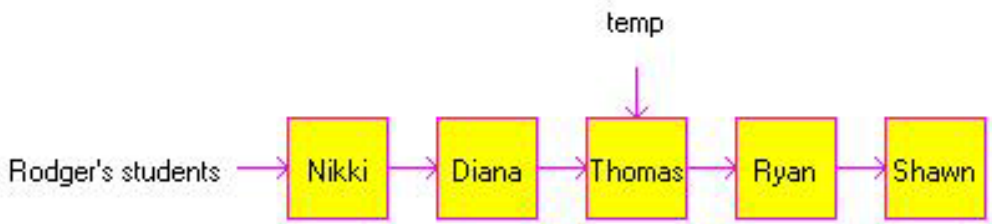
- Queue

```
queue q1 200 200 6 A 1 B 2 C 3 red blue
dequeue q1
dequeue q2
```

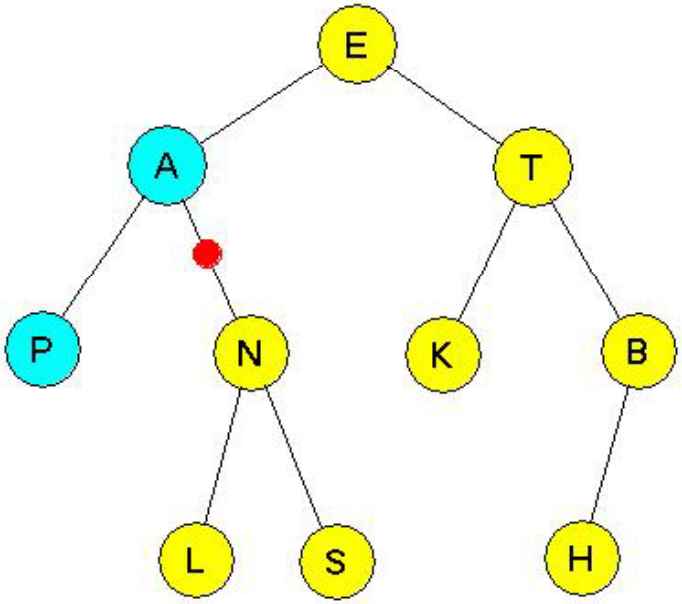


JAWAA Data Structures

- Linked List

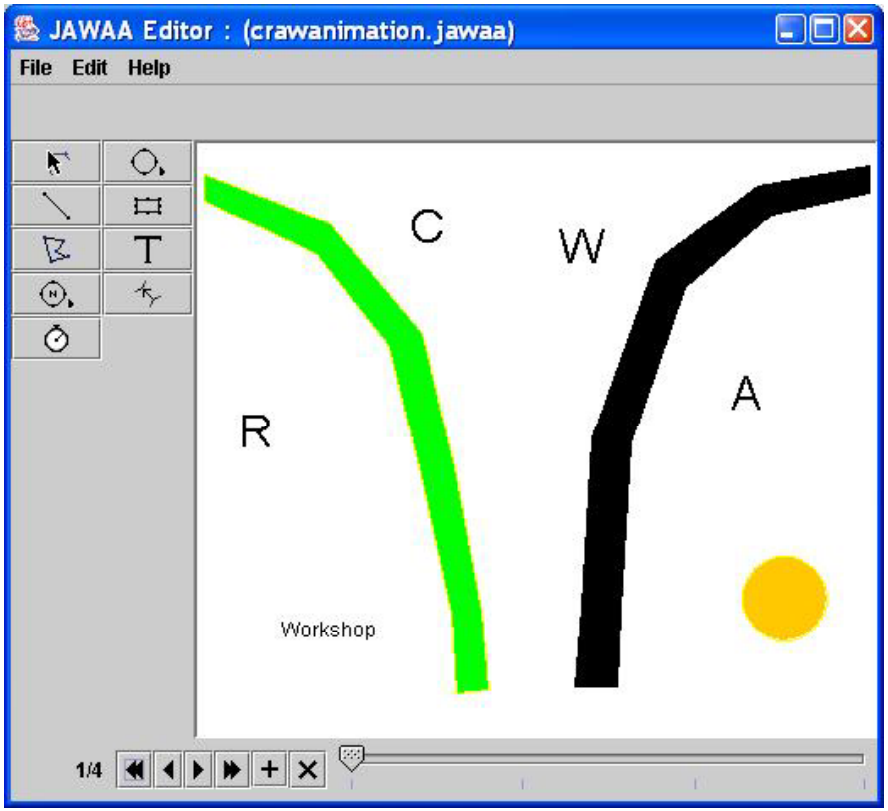


- Trees



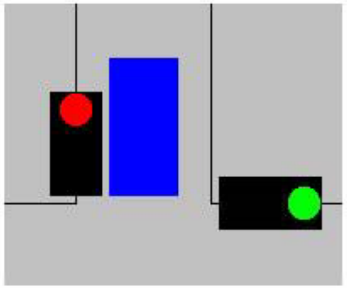
JAWAA Editor

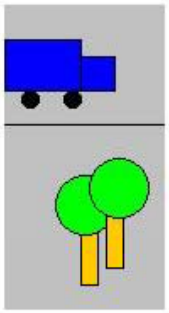
- Easily create animations
- Graphically layout primitives
- Modify across time
- No knowledge of JAWAA
- Export to JAWAA file
- Start with JAWAA editor, finish with JAWAA output from program



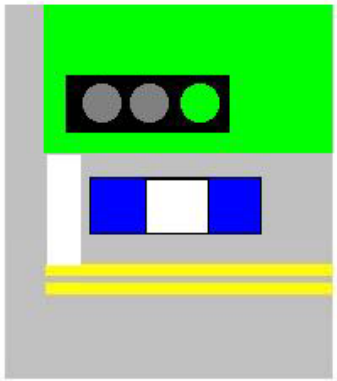
JAWAA w/o Editor vs Editor
Nonmajors course

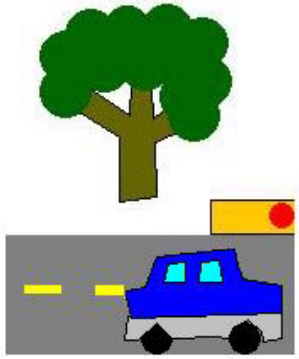
Spring 2001
No JAWAA Editor





Fall 2002
Using JAWAA Editor





Instructor Use of JAWAA in CS 1/2

- Use JAWAA Editor to make quick animations for lecture
  - Fast - 4-8 minutes each animations, Fall 2002 CS 2 Course
- Create quick animation of data structure in an existing program, add JAWAA commands as output
- Show web pages with JAWAA animations in lecture
- Students replay animations later

Instructor Animations for CS 2 Lecture

- How Pointers Work in Memory
- Recursion
- Shellsort
- Linked List - Insert at the Front
- Quadratic Collision Resolution
- Build Heap and Heapsort

Engaging Middle School Teachers and Students with Alice in a Diverse Set of Subjects

Supported by the National Science Foundation Collaborative Grant ESI-0624642, 0624654, 0624528, NSF Supplement DRL-0826661, two CRA distributed mentor awards, and two Faculty Awards from International Business Machines.

Thanks to my CoAuthors

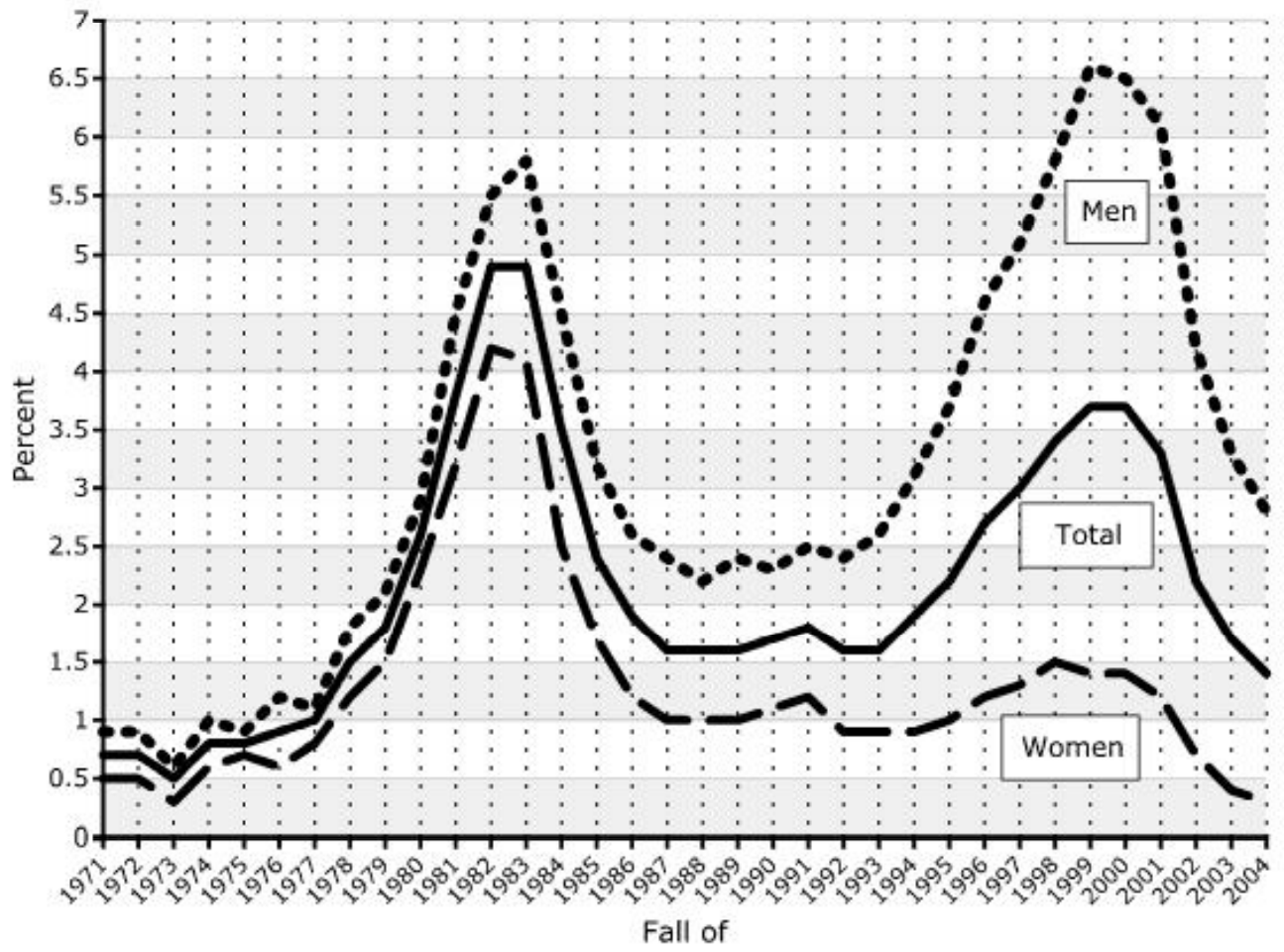
Henry Qin, Gaetjens Lezin, Jenna Hayes, Ruthie Tucker, Debra Nelson, Wanda Dann, Steve Cooper, Mercedes Lopez and Don Slater

Alice Outline

- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Alice Examples and Lesson Plans
- Usage of Alice by Middle School Students
- Summary and Future Plans

Computer Science Declining Enrollments, Few Women

Figure 1. Computer Science Listed as Probable Major Among Incoming Freshmen
Source: HERI at UCLA



How do we Introduce and Teach Science?

- Physics – experiments

- Chemistry - experiments

- Biology - experiments

We don’t introduce Computer Science in K-12!

- Not taught in middle schools and many high schools
- Students don’t know what computer science is!
- What they think it is:
  - “keyboarding, spread sheets, word processing….”
- VERY EXCITING …… NOT!

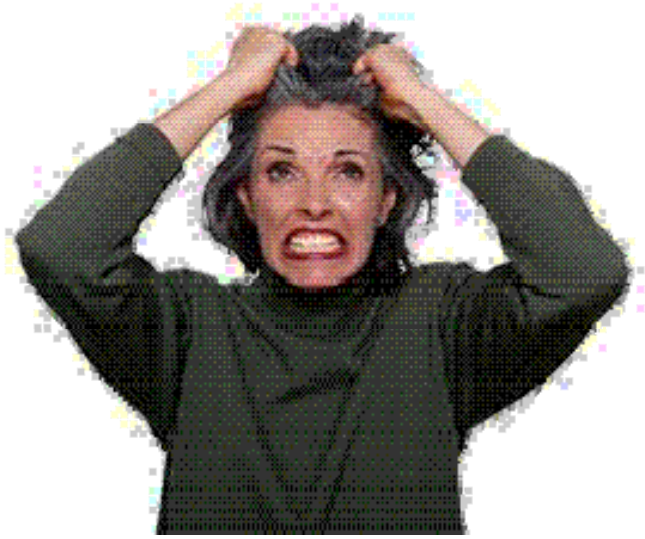
If taught, how do we introduce CS?

```
public class Simple {

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

    }

}
```

- Write a calculator
- Write a banking program
- Etc…

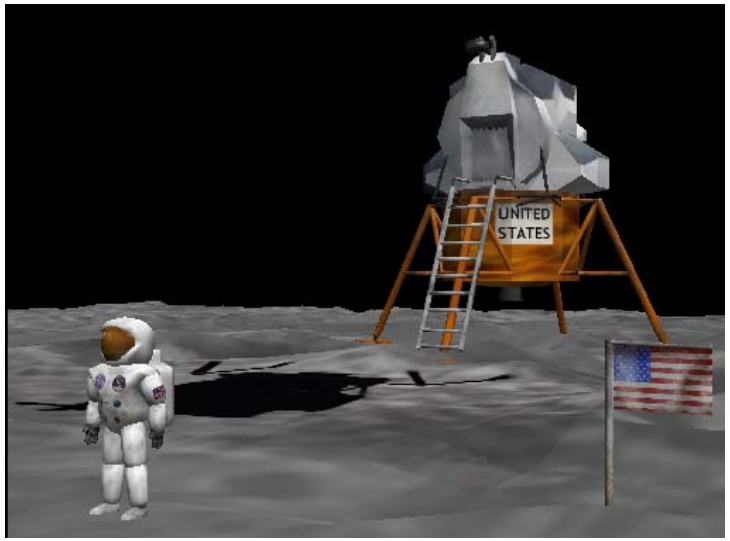
Why Can't the Introduction of Computer Science be exciting?

- Programming – it's always been
  - Hands-on
  - Interactive
  - Frustrating!
- What's missing?
  - Not Getting Exciting Results
    - Easily, right away
  - Too textual-based, including errors
  - Not appealing to today's kids in which media and technology are a part of their life!



Bring on Alice Virtual Worlds!

- Alice is
  - Hands-on!
  - Interactive!
  - Visual!
  - Less Error prone
  - Exciting Results right away!



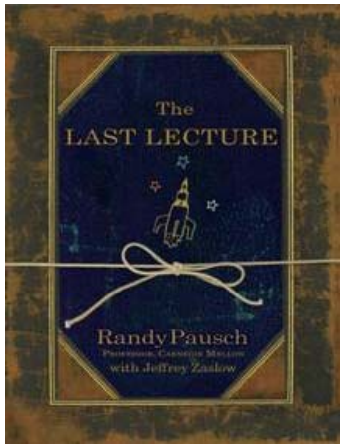
- Alice has the potential to excite kids about computer science in the same way that experiments excite kids about chemistry, physics and biology!

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
- 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



The Alice Team – Alice is free!
www.alice.org

Alice

An Educational Software that teaches students computer programming in a 3D environment

FREE!!

About Alice | Downloads | Teaching | Community | Publications | Support

Meet the Alice Team

The Alice Project is a multi-university initiative, and the Alice Team is a collaboration among faculty, staff and students.

Director

Wanda Dann

Carnegie Mellon

Contact Wanda

Research Scientist

Dennis Cosgrove

Carnegie Mellon

Homepage

Faculty

Steve Cooper

Alice Demo: Kitty Story – children's book on handicapped child

Let's visit Little Kitty the kitty. She lives with her Daddy, her Mommy, and her sister, Moon Song.

Sometimes Her mom takes her to the Doctor so that she can check out her knee. Sometimes that hurts a bit and sometimes it doesn't.

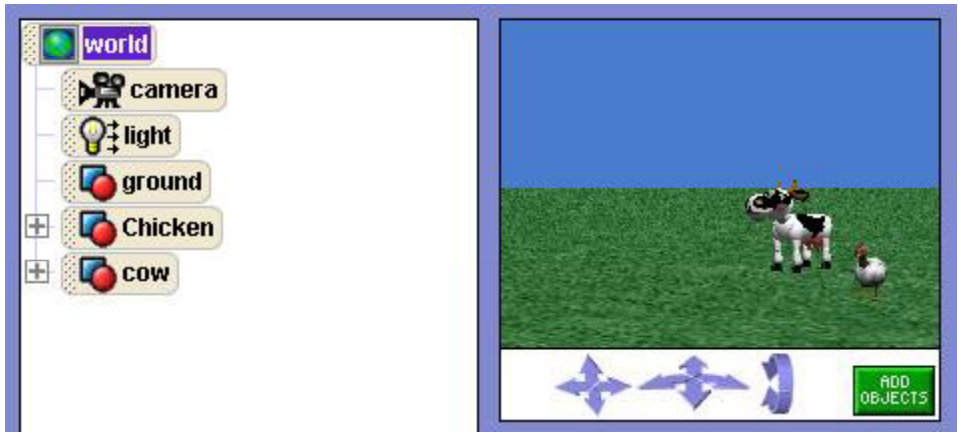
At night, her mom or dad puts leg splints on her knees. Kitty does not like this one little bit! She does a good job of crying

More on “What is Alice?”

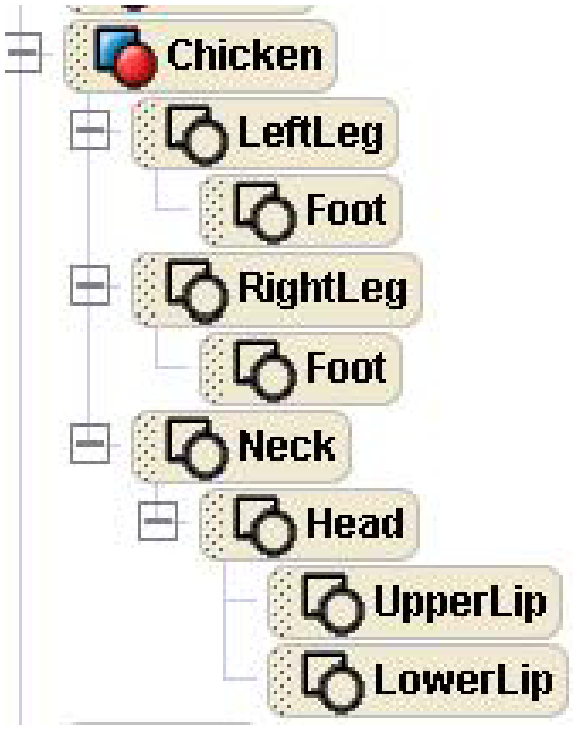
Alice Programming Language

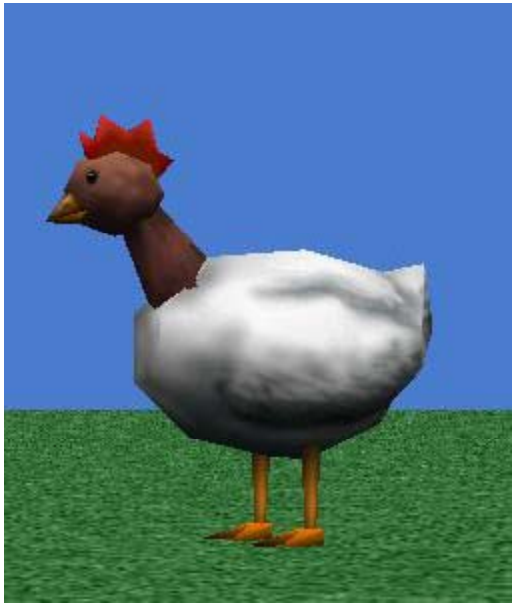
- Has libraries of 3D objects

- Keeps Track of objects you select



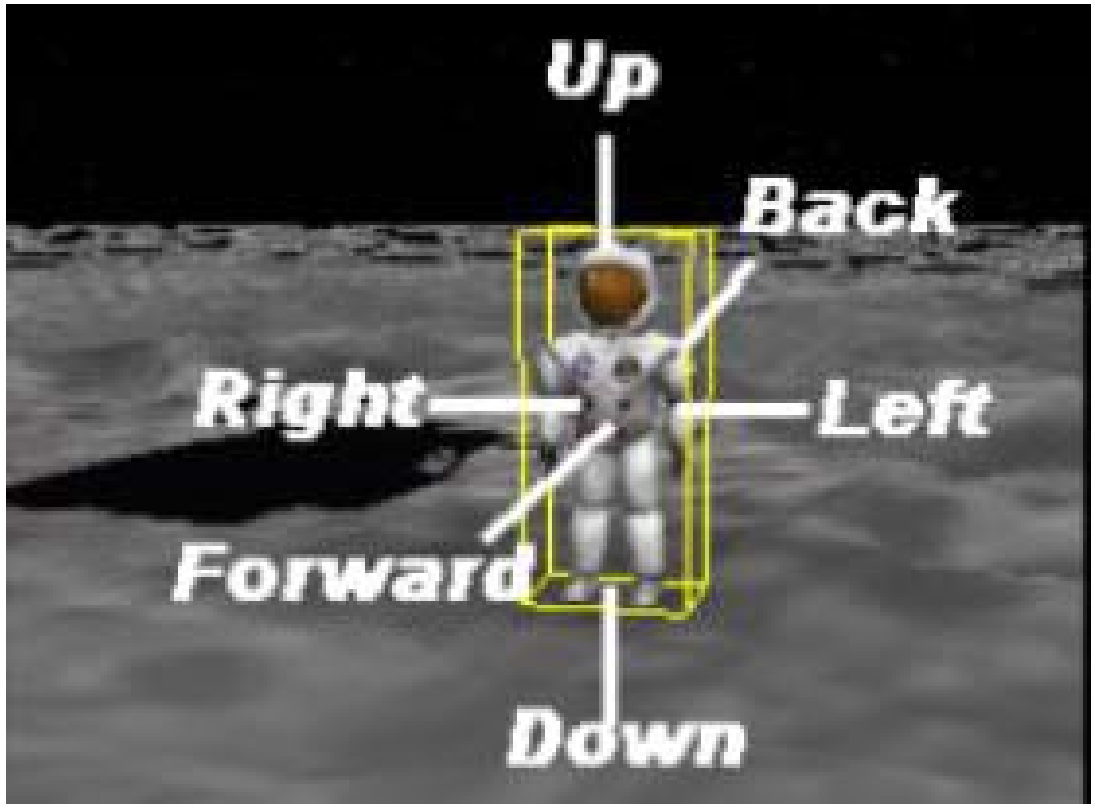
Objects Have Multiple Parts that are moveable





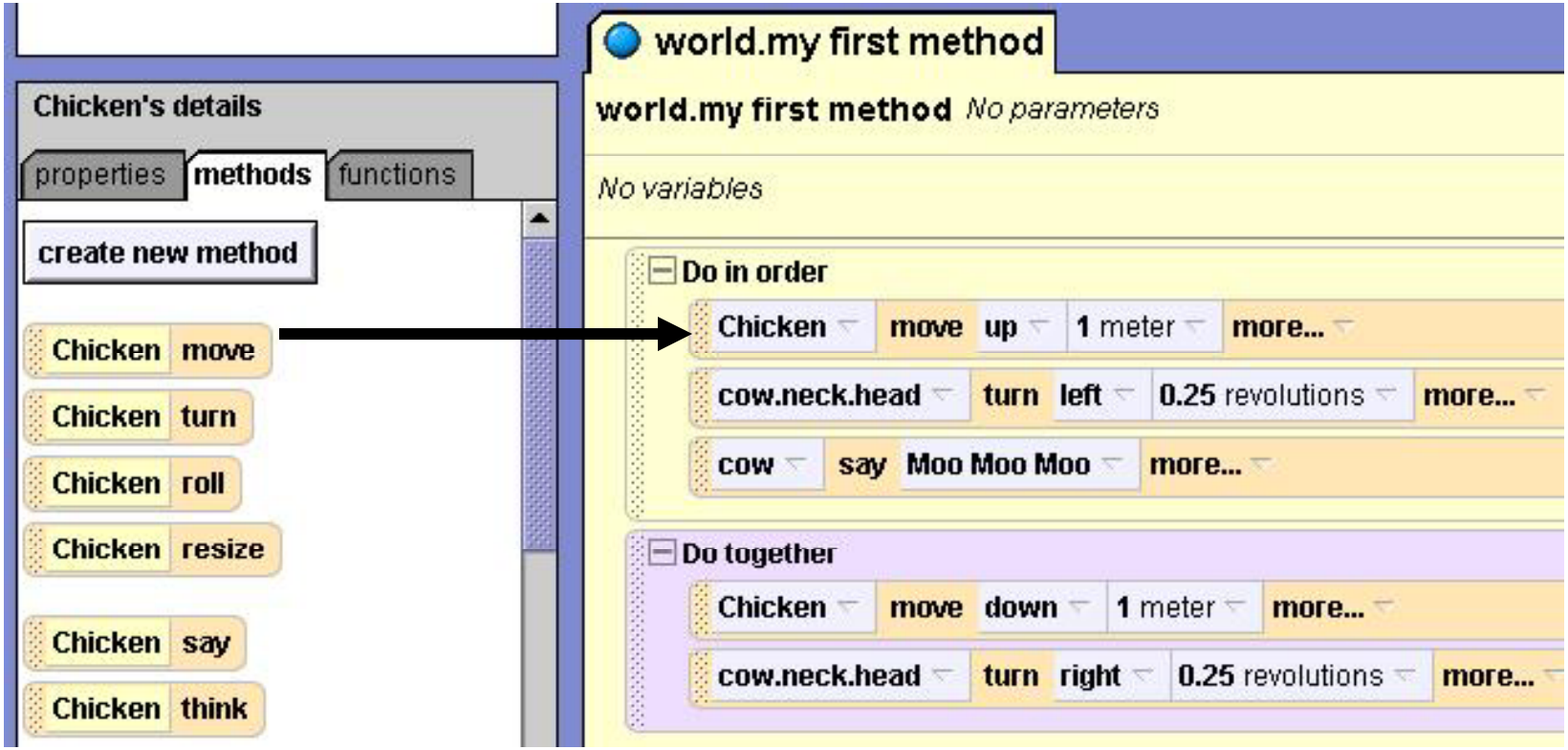
Object Position

- 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

Versions of Alice

- Alice 2.2
  - Good for Middle School/High School introduction to programming
  - Supported, will be around for awhile
- Alice 3
  - Good for full High School programming course to lead into a Java course
  - NOT READY – ROUGH BETA VERSION NOW
- StoryTelling Alice - Caitlin Kelleher
  - Written as prototype, not supported
  - PhD Thesis under Pausch

CompSci 4 – Alice Class at Duke

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

Success - Alice attracts diverse group

- At Duke
  - CompSci 4 Spring 2005
    - 22 preregister, 30 enroll (12 female + 3 African Amer.)
  - CompSci 4 Fall 2005
    - 20 preregister, 31 enroll (17 female – 1 African Amer.)
  - CompSci 4 Fall 2006 – 2 sections
    - 64 students, 33 female, 7 African Amer.
  - CompSci 4 Fall 2007 – 2 sections
    - 84 students - > 50% female
  - CompSci 4 Fall 2008 – 2 sections
    - 100 students - > 50% female
  - Advertised in school paper
    - picture of ice skater
    - Web site of animations

Games Created by Duke CompSci 4 Students

- Non-majors
- Most never programmed before
- Final projects after 10 weeks of Alice
- 50% of students are women
- Spring 05, Fall 05, Fall 06, Fall 07, Fall 08

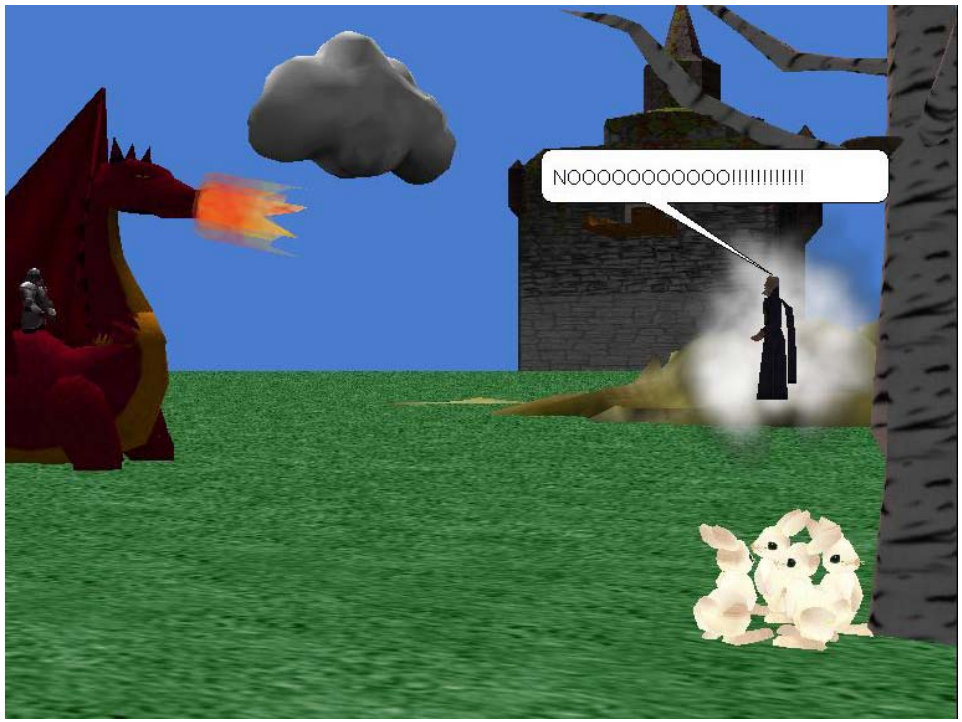
Game: Candyland

Select girl and boy to play

Click on red and green buttons to move them.

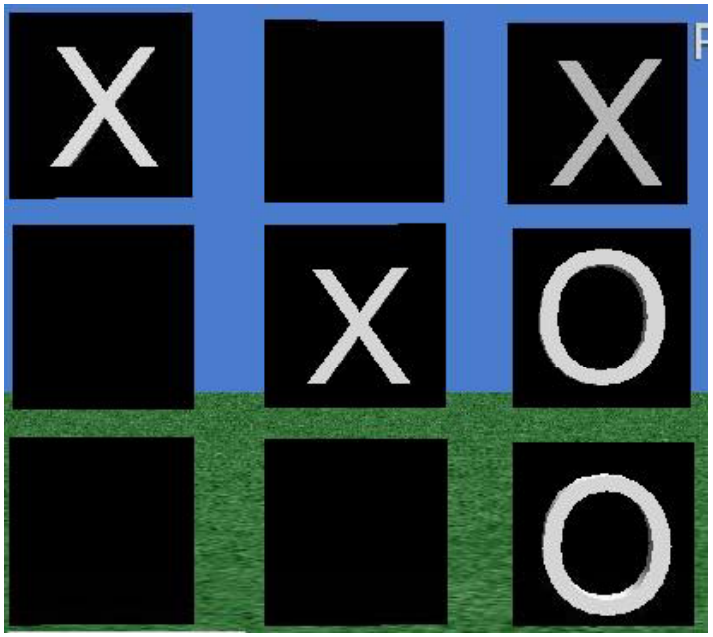
Game: Frogger – Get frog across road

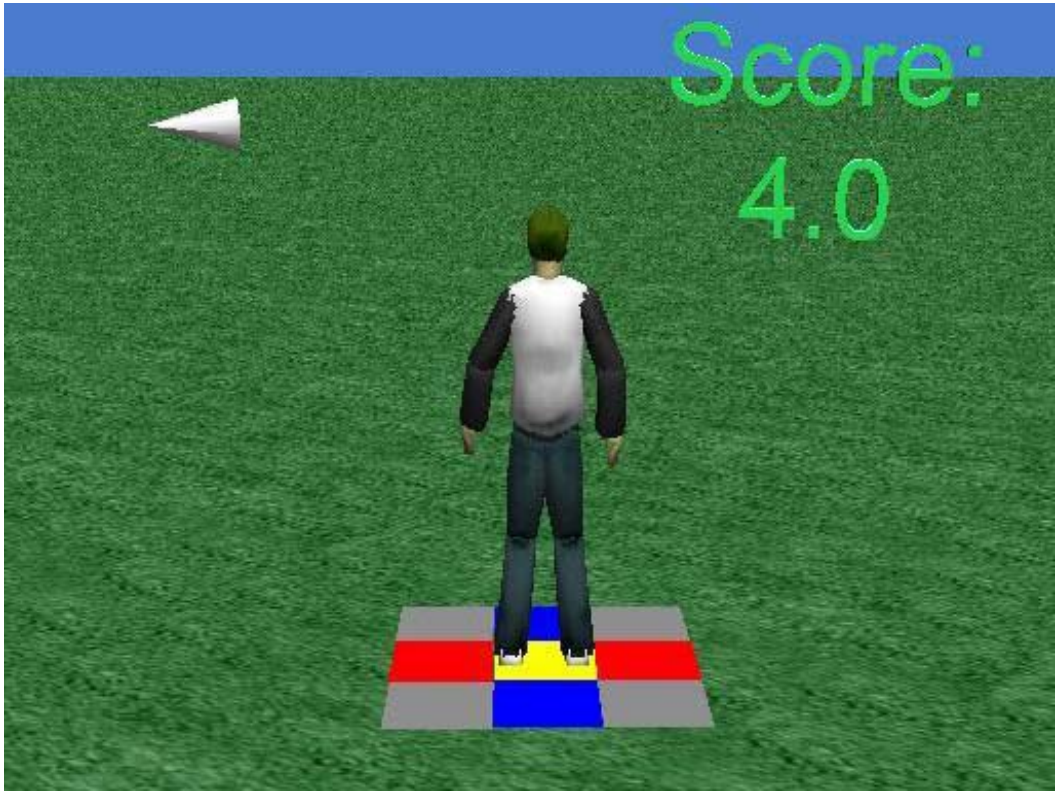
Game: Eragon



4 tasks to win the game

Game:
Tic Tac Toe

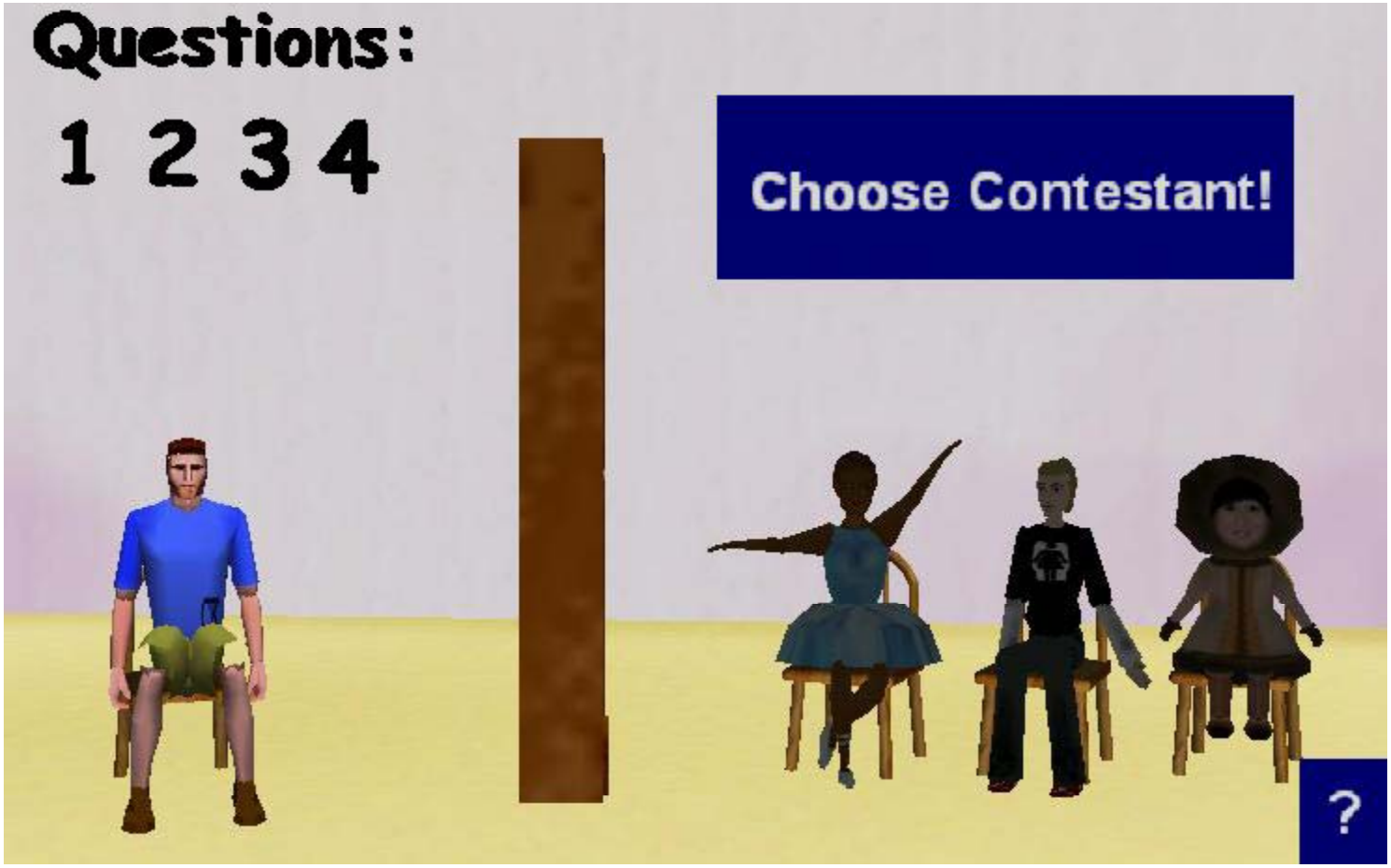




Game: DDR

Click on arrow keys,
Player moves foot to square

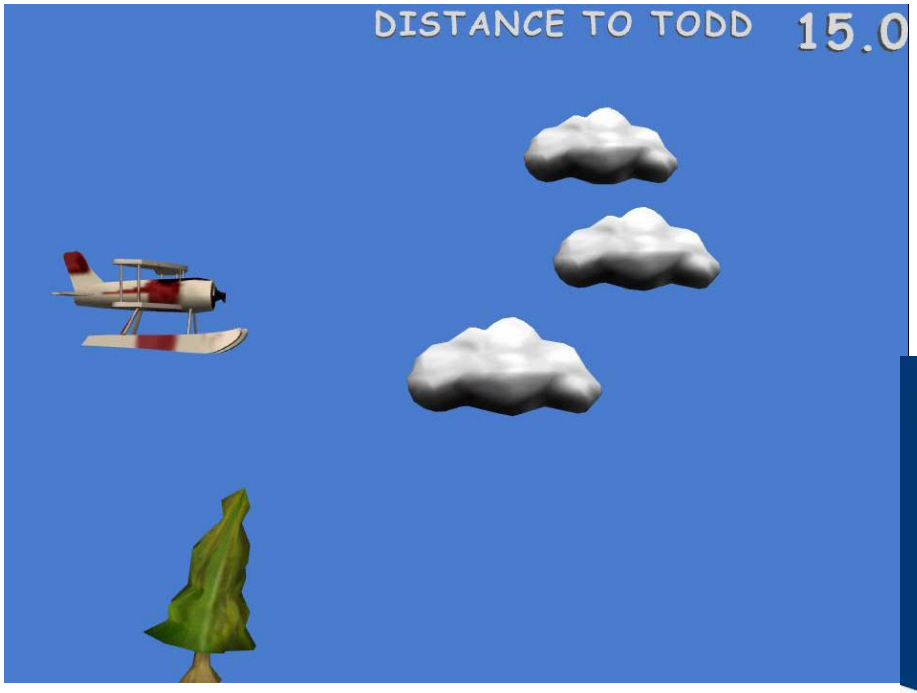
Game: Dating Game



Game:
Rumble Putt

Game: Sarah Palin’s Seaplane Adventure

Sarah Palin’s Seaplane Adventure (cont)



Outline

- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Alice Examples and Lesson Plans
- Usage of Alice by Middle School Students
- Summary and Future Plans

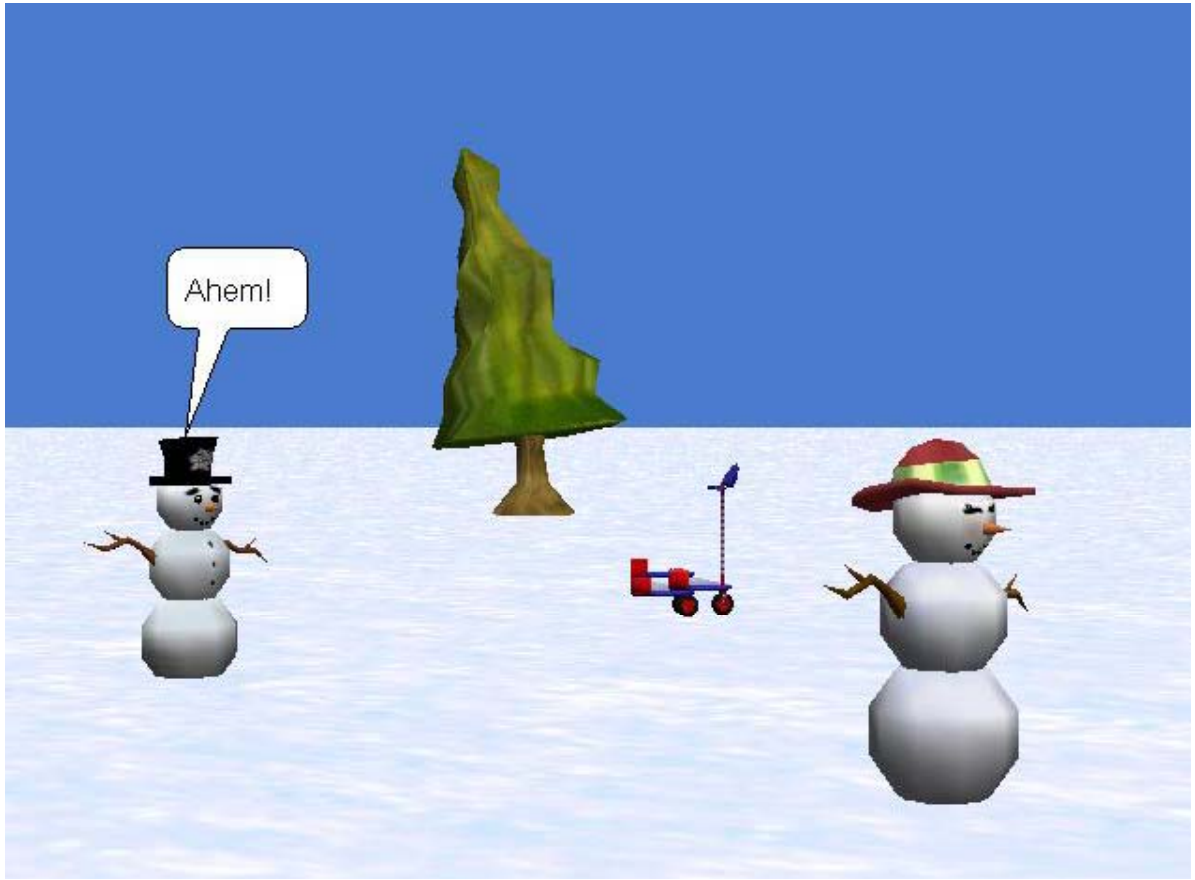
Transition to K-12

Alice into K-12

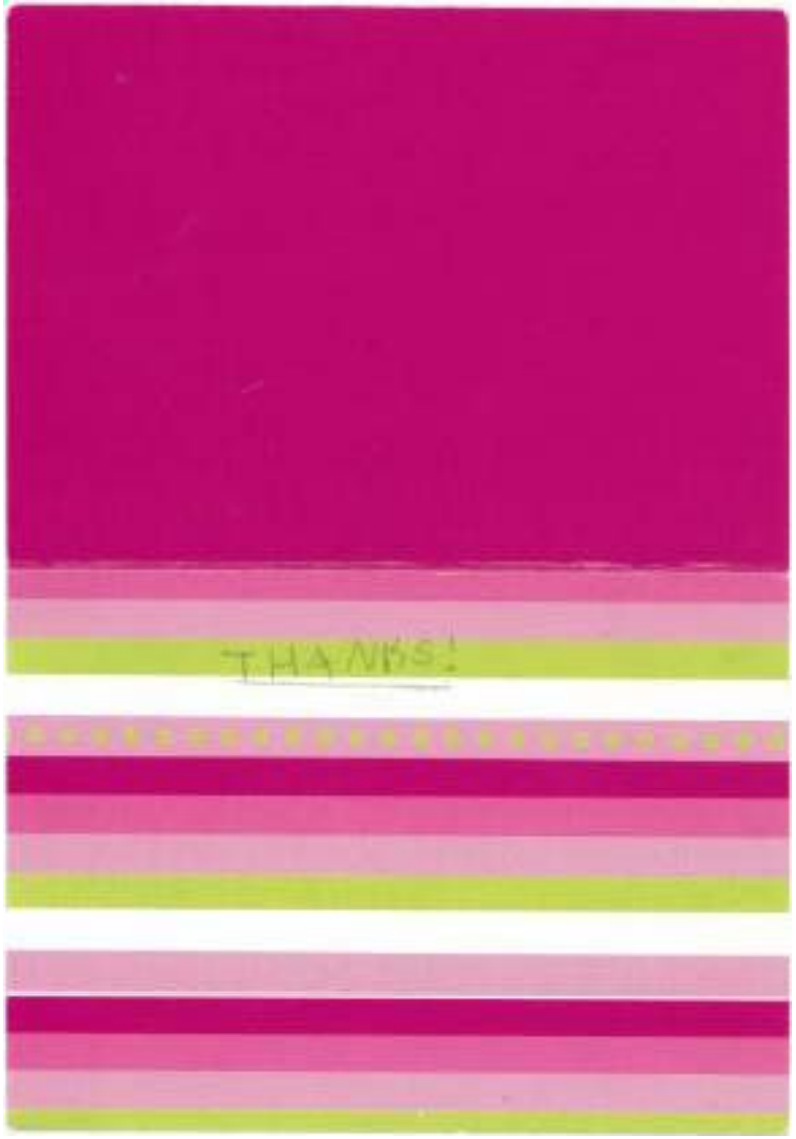
- Non-majors course at Duke
  - Popular, fills up with seniors
  - College students pretty set with their major before they come
- Students in middle school are starting to form decisions on careers
- They have exposure to Teachers, Doctors, Astronauts, etc.
  - BUT DON’T KNOW WHAT COMPUTER SCIENCE IS

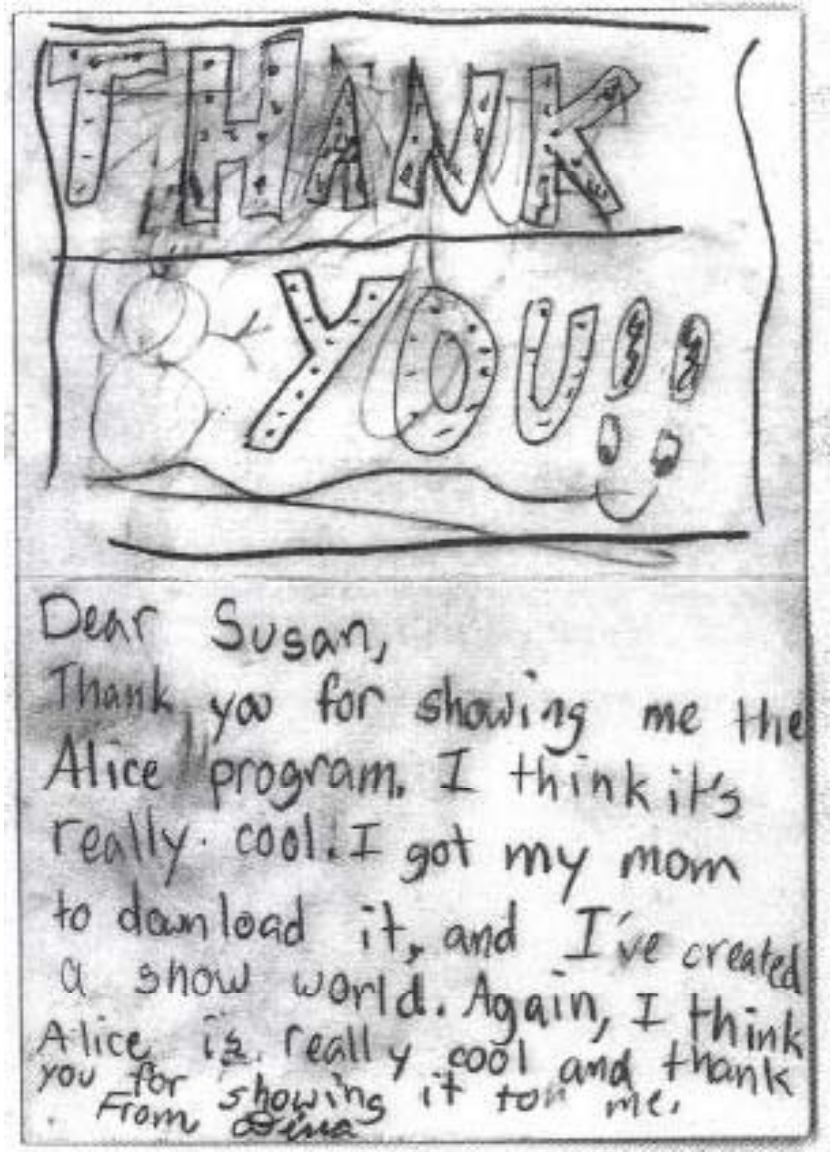
Success - Alice Excites 4th-6th Grade Girls

- Duke Femmes Event, April 07
- 60 girls – 4 groups of 15
- Taught them Alice for an hour
- Handout to take home
- Event again in 2008 and 2009



Thank you from 4th Grade Girl





Adventures in Alice Programming
Grades 5-12 Outreach

www.cs.duke.edu/csed/alice/aliceInSchools

Adventures in Alice Programming

- Integrate Alice into high school and middle schools by training teachers
- Six sites:

Durham, NC	Charleston, SC	Virginia Beach, VA
Denver, CO	Oxford, MS	San Jose, CA

- Durham site focuses on Middle Schools in NC

www.cs.duke.edu/csed/alice/aliceInSchools

Duke: Adventures in Alice site

- Summer 2008
  - 3-week Teacher workshop
    - 35 teachers, mostly middle school, some high school
    - Only a few had ever programmed before
    - Subjects: english, math, science, history, art, technology
    - Taught them Alice, Developed Lesson Plans
  - Two one-week middle school camps
    - Taught Alice
    - Lots of time to build their own Alice worlds
  - Overlap between the two
  - Followup Teacher workshop Summer 09

How to Use Alice in Middle Schools

- Teachers
  - Examples in lecture
  - Make interactive quizzes
  - Make worlds on concepts for students to view
- Students
  - Projects (in place of a poster, a model)
  - To take quizzes
  - To view and answer questions about a world

Outline

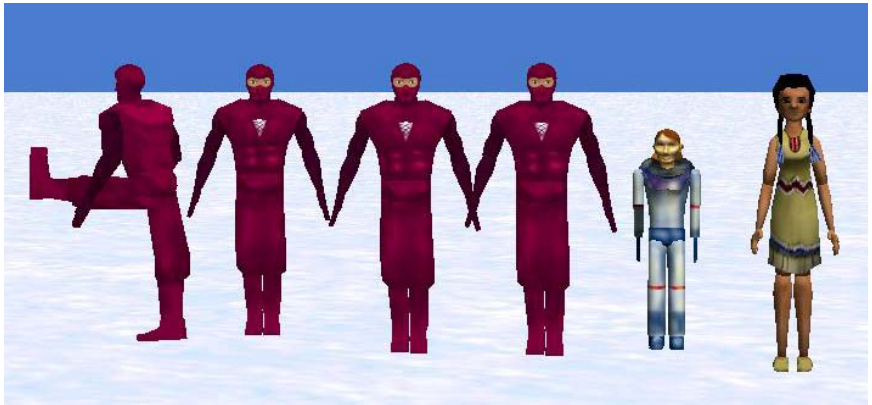
- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Alice Examples and Lesson Plans
- Usage of Alice by Middle School Students
- Future Plans

Three Introductory Tutorials

1. Simple, Short (15 min) tutorial to try Alice
   - Add an object, use built-in methods
2. One hour tutorial for younger kids
   - Writing methods, simple events, camera
3. Four part tutorial for middle school kids
   - More detailed on placement of objects, writing methods, events, camera control
   - How to put a person on a horse
   - Answer a cell phone

Many short tutorials on CS Topics

- CS Topics
  - Programming – sequential and “at the same time”
  - Methods
  - Events
  - Looping
  - Conditionals (making a choice)
  - Functions (compute and return an answer)
  - Lists
  - Variables (timers/scores)



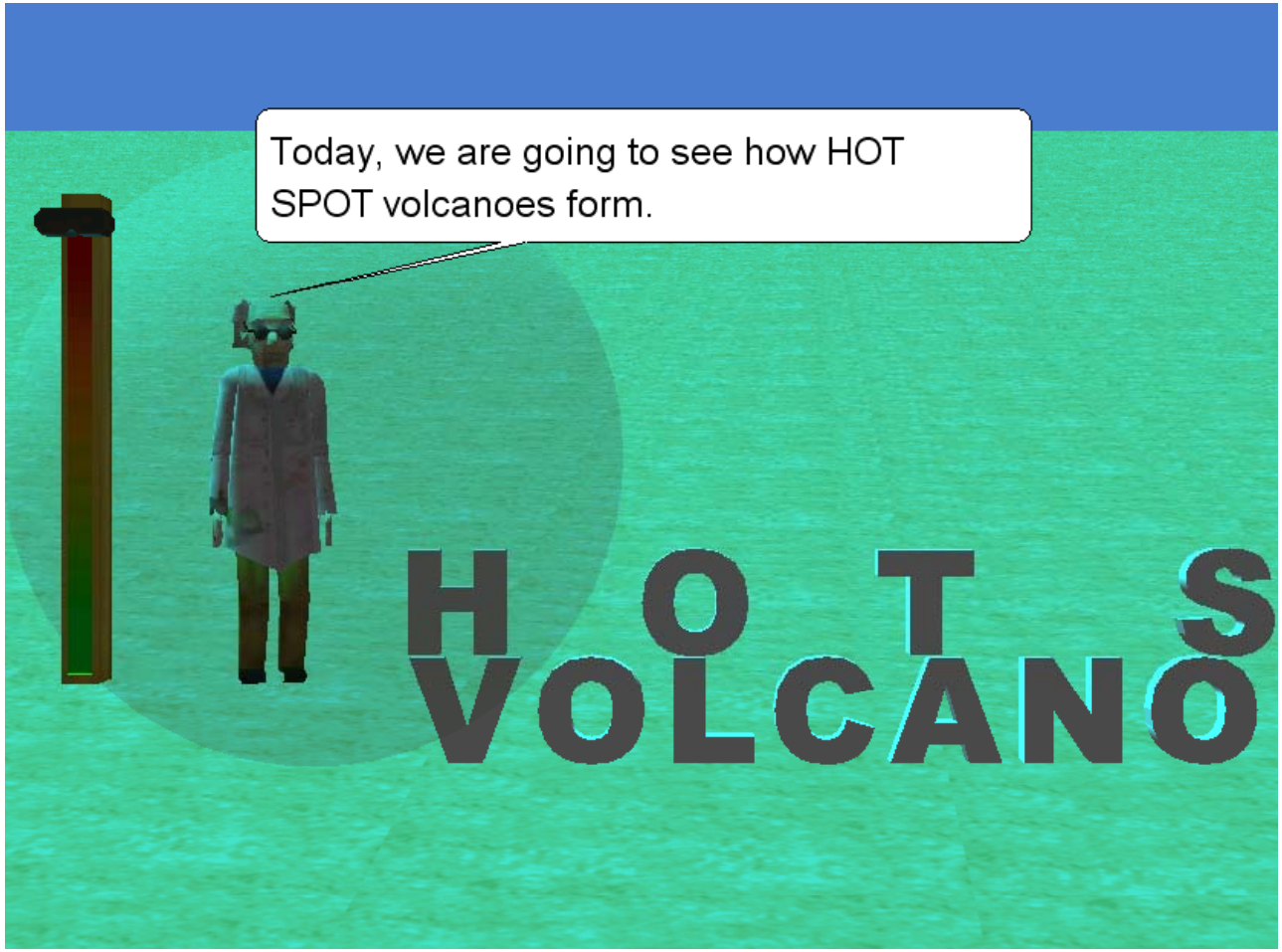
Other “Fun” Topics Blended in

- Storyboards
- Changing camera views
- Scene changes and lighting
- Making Billboards
- Making objects invisible and visible
- Sounds
- Glueing objects to others

Outline

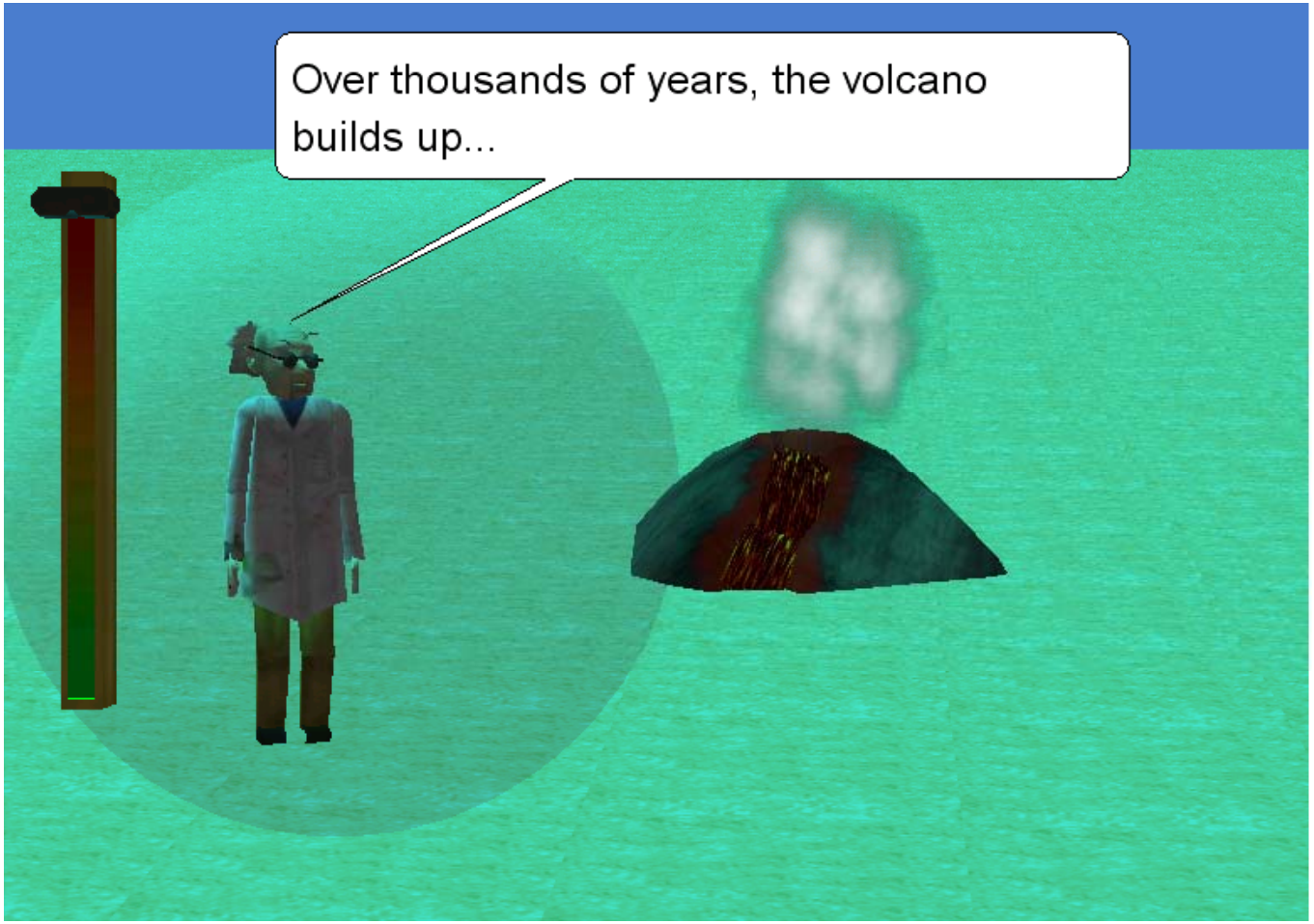
- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Examples and Alice Lesson Plans
- Usage of Alice by Middle School Students
- Summary and Future Plans

Science Example:
How volcano is formed

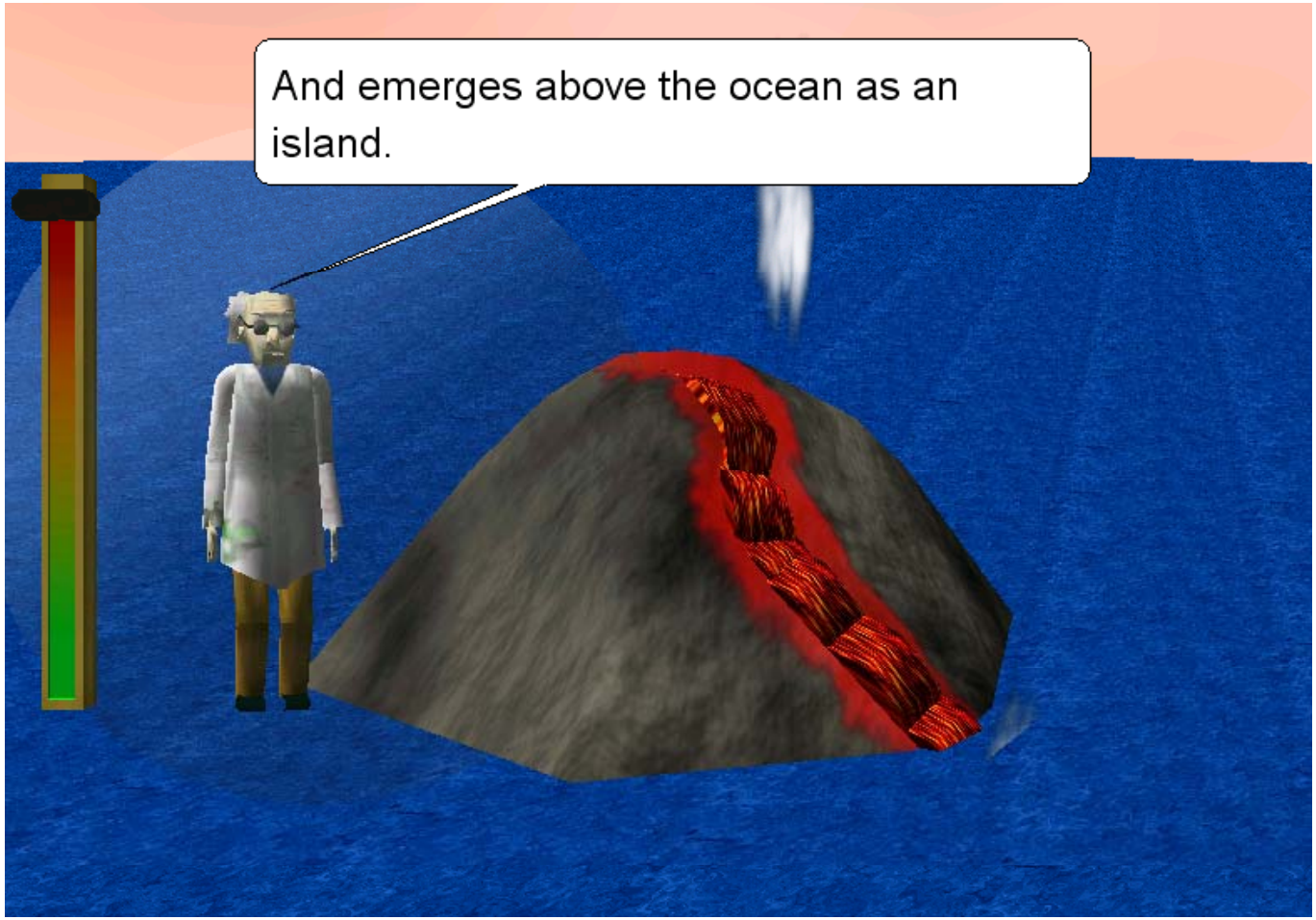


How a volcano is formed (slide 2)

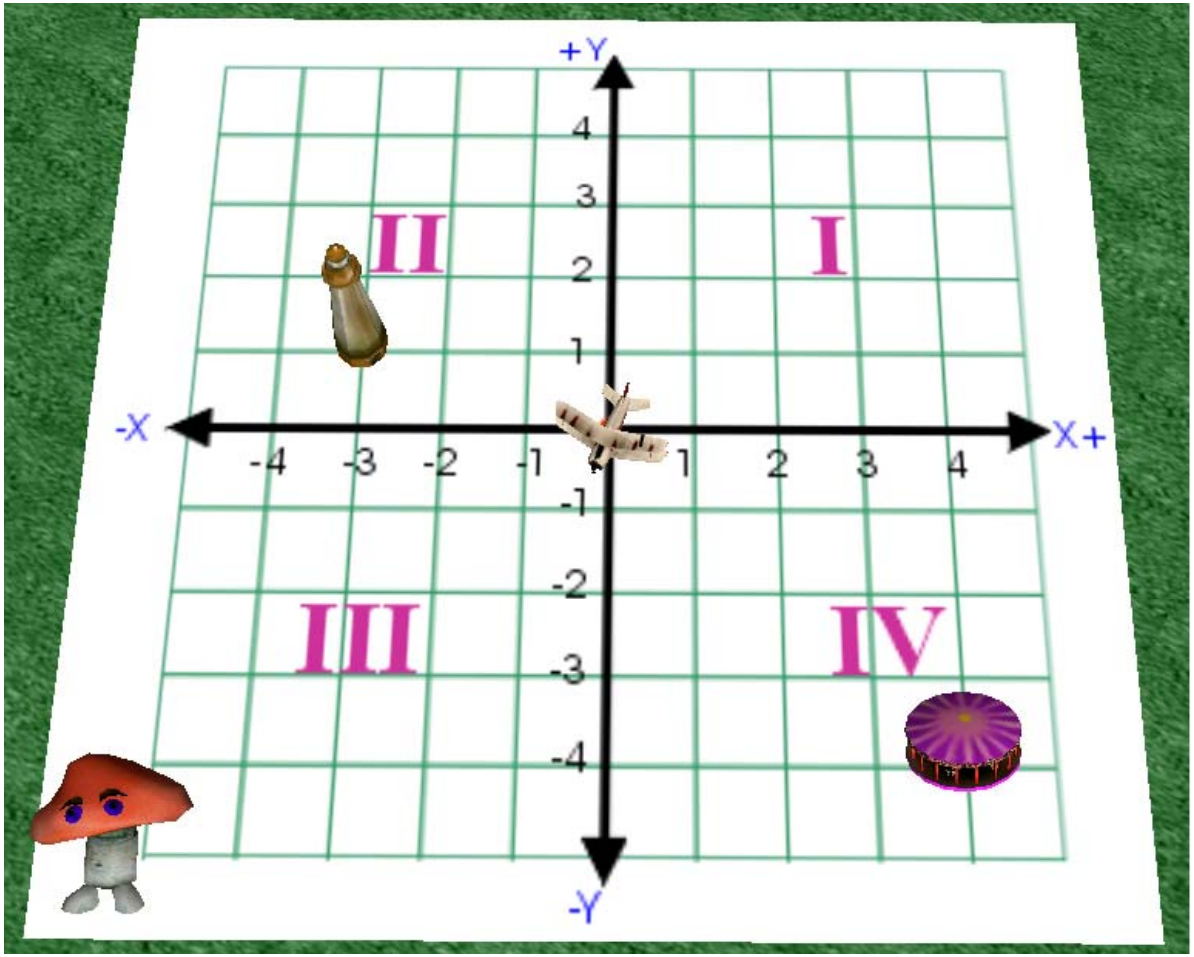
How a volcano is formed (slide 3)



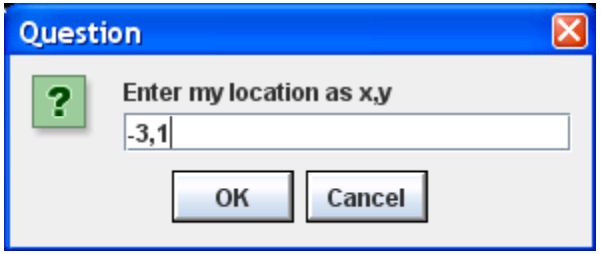
How a volcano is formed (slide 4)



Math Example:
Teacher Lesson Plan on quadrant plane



- Click on lighthouse
- Enter x,y position
- Objects randomly move



Magic Tree House Quiz
Famous Children's Book Series

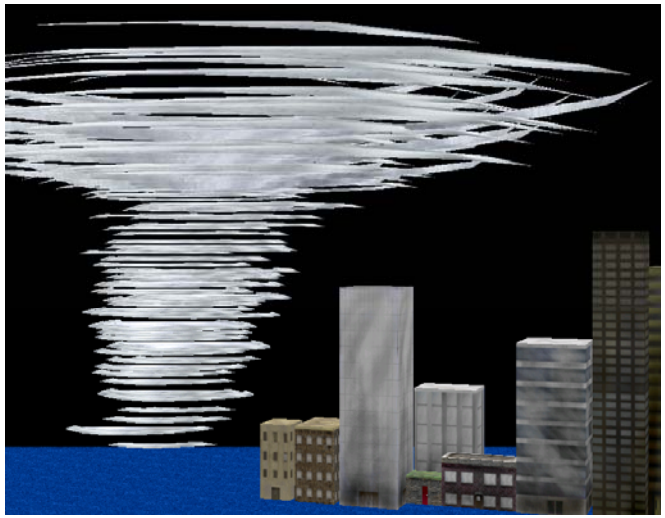
Other Ideas for Projects

- Story from Ancient Egypt
- Spanish Quiz in which you see a word and have to click on the object the word represents
- Animate a scene from a book you have read or a poem you have written
- Create a world about school safety
- Memory game – remember a random color sequence
- Math Quiz – Answer the questions

Alice worlds for these and more are on our website.

Other Teacher Lesson Plans

- Math
  - Finding surface area
  - Rate of Change and Slope
- Science
  - Create a food chain
  - Sun, Earth and Moon system
  - Tornados
  - Physics – Newton's law of gravity
  - Alternative Energy



Other Teacher Lesson Plans (cont)

- History/Social Studies
  - The continents – view world and answer questions
  - Animated overview of Japan
  - Animated overview of Egypt

- English
  - Write and animate a poem
  - Animate a poem or scene from a story
  - Write a movie trailer

Outline

- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Alice Examples and Lesson Plans
- Usage of Alice by Middle School Students
- Summary and Future Plans

What type of objects did they use?

- Girls top five
  - People, animals, environments, nature, 3D-text
- Boys top five
  - Vehicles, people, buildings, scifi, special effects

Typical Boy Example

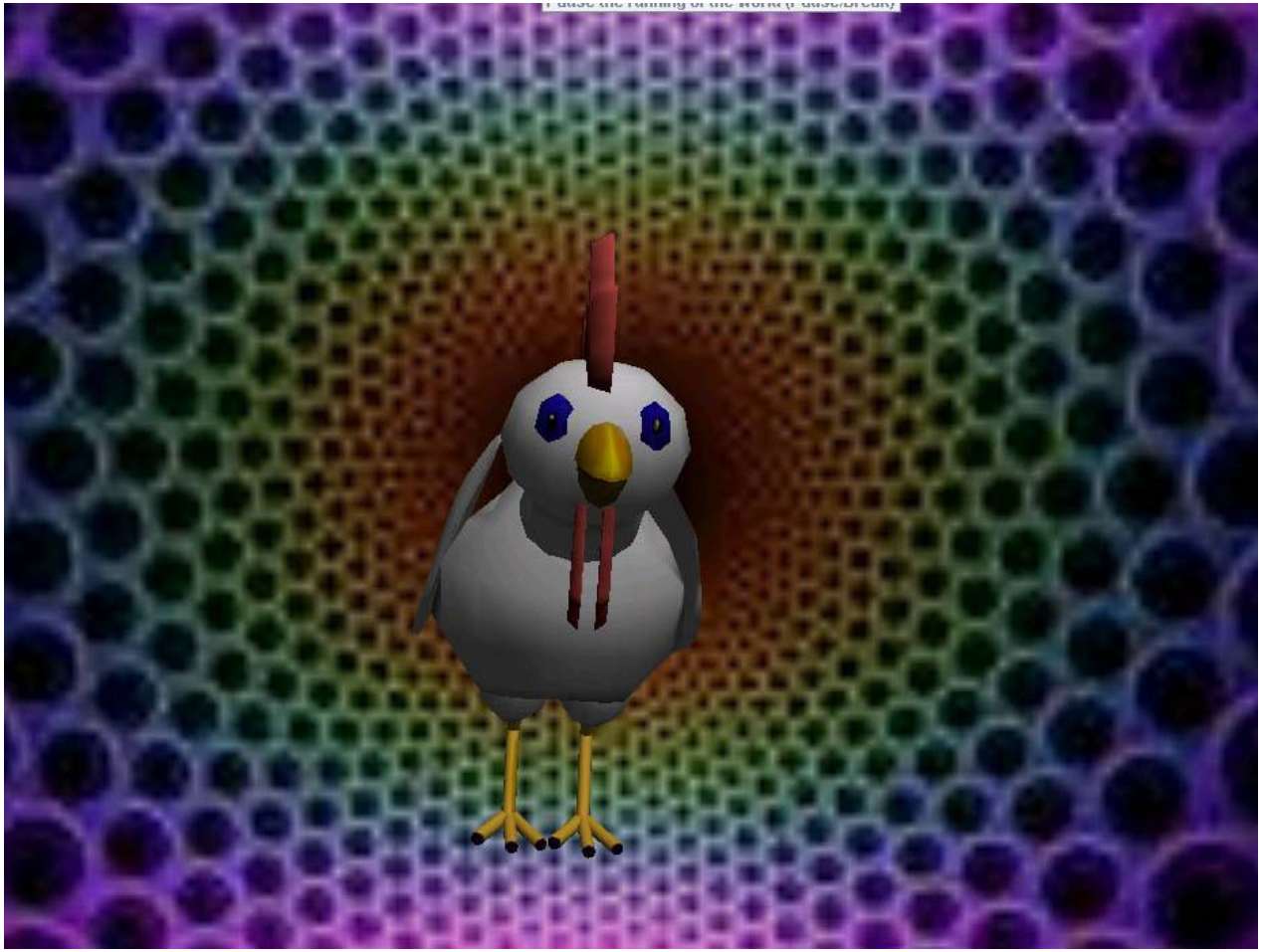
SciFi, vehicles, fire

More fire

And more fire

And more fire!

Girl Examples – Dancing chicken



Girl Example 2 - Egypt

Girl Example 3 – Attack of the lemurs

Girl Example 4 - carnival

Girl Example 5 – rescue baby

How did the Students use Alice?

- Examined worlds to see which concepts they used

TOPIC	at least once	3+ times	
parameters	34%	17%	CS Topics
loop	57%	23%	
list	45%	8%	
simple event	57%	34%	
4 arrow event	60%	26%	
if statement	43%	11%	
vehicle property	88%	46%	Basic topics
camera controls	80%	51%	
scene change	51%	26%	
color property	66%	17%	

Feedback from Parents

- “[My daughter] thoroughly enjoyed her week with you. It was a great experience!”
- “I’m convinced. Kids like Alice and Alice is a good way to teach kids programming. [My son] is doing my python course and he’s not all that interested in python and never touches it between the courses. However, in the evenings when he comes home from the Alice course, he works on his Alice worlds.”

Outline

- Motivation and background
- Adventures in Alice Programming overview
- Middle School Alice Tutorials
- Middle School Alice Examples and Lesson Plans
- Usage of Alice by Middle School Students
- Summary and Future Plans

Summarizing

- We developed
  - Tutorials
  - Examples of possible use in Middle Schools
- Teachers developed
  - Lesson Plans for history, science, math, language arts, art, and technology
  - Animation Fair
- Middle School Students
  - Were engaged, developed their own worlds
  - Animation Fair
  - Difficult to get away from the computer

All materials are on our website.

Alice Symposium and workshops in 2009

- June 17, 2009 – Alice Symposium
  - Submit papers by March 15th
  - www.cs.duke.edu/csed/aliceSymposium2009/
- Three one-week Alice workshops
  - June 22-26
  - June 28- July 2
  - July 6-10
- Two day followup Alice workshop
  - June 15-16

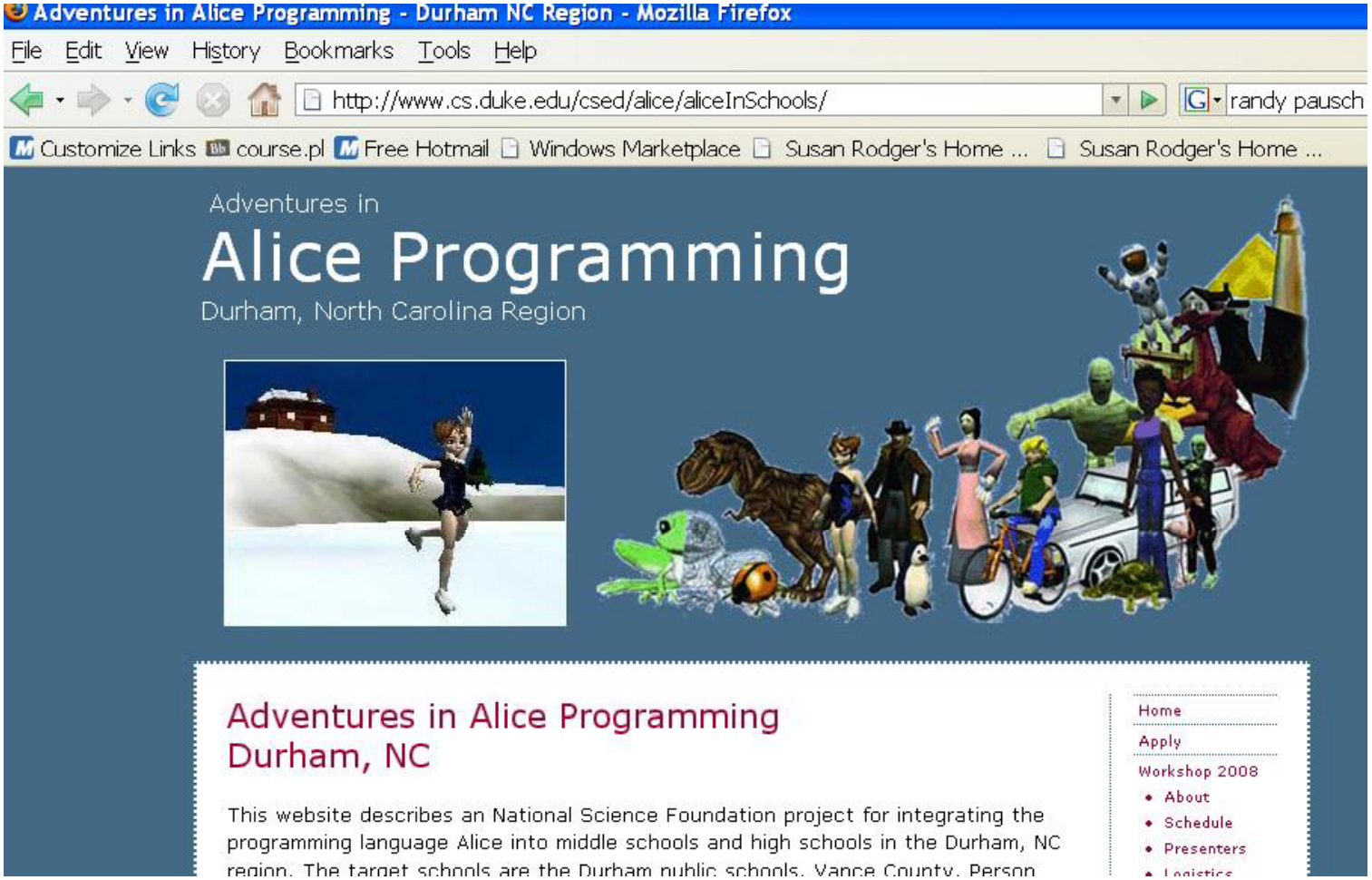
Results of our workshop this summer

- Teachers are very excited about Alice
- Teachers want many specific models built
- We are developing classes that could be helpful to teachers
  - Quiz class
  - Timer and Score class
  - Super ground class

Web site

- Adventures in Alice Programming

  www.cs.duke.edu/csed/alice/aliceInSchools



Questions?